aerospace

MDPI

Article

Detecting Satellite Breakup Events via Density-Based Spatial Clustering of Space Debris

Jing-Hui Zheng *, Yin-Dun Mao, Juan Shi and Qi Liu

Shanghai Astronomical Observatory, Chinese Academy of Sciences, NanDan Rd. No.80, Shanghai 200030, China
* Correspondence: zhengjinghui@shao.ac.cn; Tel.: +86-17610065902

Abstract

When artificial satellites disintegrate in orbit due to collisions, explosions, or anti-satellite tests, they generate a large amount of space debris. These debris move in orbit through formation-like flight patterns, which we refer to as debris clusters. Compared to constellation satellites in formation flight, debris clusters have closer mutual distances and higher spatial density. To address this phenomenon, this paper proposes a detection method for satellite breakup events through a two-layer clustering strategy using the DBSCAN algorithm in three-dimensional spatiotemporal domains. The method provides a systematic approach for identifying and characterizing breakup debris through orbital backpropagation and density analysis, serving as a screening tool for potential breakup events. Using the 2009 Cosmos 2251-Iridium 33 collision breakup event as a validation case, experimental results demonstrate that our two-layer clustering approach achieves good clustering accuracy (98.92%) and a positive recall rate (93.20%), with a temporal resolution of 5 s and spatial precision of 21.68 km. The methodology was further applied to analyze the 2024 Intelsat 33E breakup event, successfully identifying sparse debris clusters from Intelsat 33E. The detected breakup time showed an 8 min deviation from the Space-track.org official report.

Keywords: satellite breakup events; space debris; debris clusters; DBSCAN; space situational awareness

1. Introduction

The launch of the inaugural artificial satellite by the Soviet Union in 1957 marked humanity's entry into the space age and the beginning of near-Earth space exploitation. The advancement of aerospace technology has led to a dramatic increase in the number of Resident Space Objects (RSOs). European Space Agency (ESA) statistics show that as of March 2025, about 6890 launches have placed about 21,320 satellites into Earth orbit and have created over 60,000 tracked objects. About 40,360 of these objects remain in orbit and are regularly tracked and catalogued by Space Surveillance Networks [1]. The U.S. space surveillance network reports that there are currently approximately 18,800 pieces of debris in their catalog, including those larger than about 5–10 cm in Low Earth Orbit (LEO) and 30 cm to 1 m in Geosynchronous orbit (GEO) altitudes, which may lead to fatal space accidents.

The massive accumulation of space debris poses an escalating threat to spacecraft orbital safety, leading to a significant increase in the frequency of space incidents. Since the start of the space age in 1957, there have been more than 650 break-ups, explosions, collisions, or anomalous events resulting in fragmentation [2].

Check for updates

Academic Editors: Marco Felice Montaruli and Mauro Massari

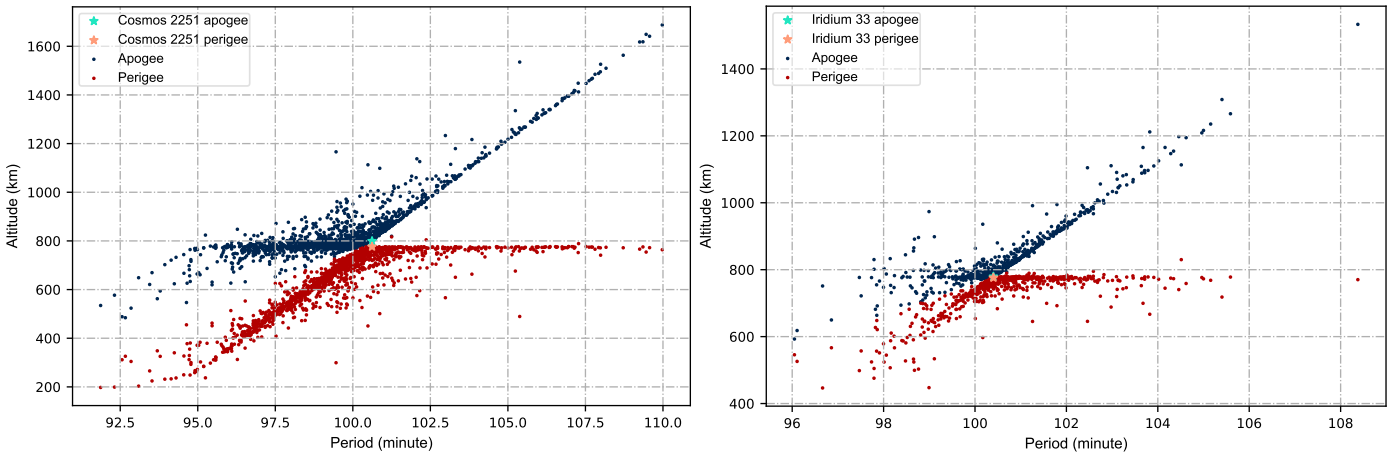
Received: 21 February 2026
Revised: 12 April 2026
Accepted: 13 April 2026
Published: 22 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

On 10 February 2009, the Russian Cosmos 2251 satellite collided with the American Iridium 33 satellite in orbit, generating over 1000 debris. The evolution of debris clusters has exerted a significant impact on the orbital environments surrounding both satellites [3]. On 19 October 2024, the Intelsat 33E satellite disintegrated in orbit. The U.S. Space-Track.org website released orbital data for 18 debris, while the Global network of optical telescopes of the Shanghai Astronomical Observatory (SHAO), detected over twenty debris pieces, and Roscosmos officially announced the detection of more than eighty debris pieces generated from this breakup event. Unlike the previous collision case, this represents a non-collision-induced natural satellite breakup event. In recent years, with the exponential growth of space objects, the frequency of satellite breakup events has significantly increased. In addition to the incident where the Intelsat 33E satellite disintegrated in orbit, multiple satellite breakup events of unknown origins successively occurred in 2024, generating substantial debris clusters.

The occurrence of these events demonstrates the rapid deterioration of traffic safety in the current near-Earth orbital environment. The increasing population of artificial satellites and space debris elevates the probability of satellite breakup events caused by collisions or explosions. The resultant debris clusters further amplify the quantity of space debris, potentially triggering Kessler Syndrome [4]. This vicious cycle leads to uncontrolled growth of space debris [5,6], rendering the near-Earth space environment increasingly critical. The detection, monitoring, and governance of space debris demand immediate action. To prevent accelerated degradation of space traffic safety, it is imperative to enhance space situational awareness capabilities and strengthen cataloging management of RSOs, enabling timely warnings for potential space accidents such as collisions. Particularly, the detection and rapid identification of satellite breakup events constitute a crucial component in this framework.

From the perspective of orbital characteristics analysis, debris generated by satellite breakup events exhibits distinct apogee–perigee distribution patterns [7]. Taking the collision and breakup event between Cosmos 2251 and Iridium 33 satellites as an example, the Gabbard diagrams of their debris at the breakup epoch are shown as Figure 1.



(**a**) (**b**)

Figure 1. Gabbard plot of satellite Cosmos 2251 debris and satellite Iridium 33 debris at 10 February 2009. (**a**) Orbit apogee and perigee of Satellite Cosmos 2251 and its debris. (**b**) Orbit apogee and perigee of Satellite Iridium 33 and its debris.

The Gabbard diagram constitutes an essential tool for analyzing breakup events through orbital characteristic distributions of satellite breakup debris. In the collision case involving Cosmos 2251 and Iridium 33 satellites, most debris fragments exhibit orbital eccentricity comparable to their parent satellites, demonstrating apogee and perigee distri-

butions as illustrated in Figure 1. However, prior to debris authentication, orbits generated by satellite breakup events may resemble those of other space objects or unidentified debris. During the immediate post-breakup phase, when target identification has not yet been initiated, the Gabbard diagram alone presents limited capability to distinguish between breakup-associated debris and other uncatalogued objects.

Most satellite breakup events exhibit characteristics of substantial debris generation and high distribution density. These debris typically demonstrate distinct spatiotemporal aggregation tendencies. Whether caused by collisions or other factors such as thermal stress, battery explosions, or residual fuel detonations, the majority of debris resulting from breakups can be traced back to the proximal space surrounding one or multiple parent objects near the breakup time. Leveraging this characteristic, we propose the design of an automated detection method for space debris clusters and the analysis of breakup events during the initial phase of satellite breakup events.

The forensic analysis of satellite breakup events has been extensively studied. Refs. [8,9] developed methodologies for parent body identification through orbital distance analysis. Ref. [10] proposed a stochastic approach to detect fragmentation epochs from single-fragment observations. For rapid event characterization, ref. [11] presented techniques for quick-response representations of fragmentation events. Recent advances include the reconstruction of long-term breakup events [12] and early-stage characterization methods [13]. Unlike the approaches focusing on detailed post-event reconstruction, our method emphasizes real-time detection and initial assessment capabilities suitable for operational space situational awareness (SSA) systems.

In operational space object surveillance practice, the detection of numerous debris from breakup events typically manifests the following phenomena:

1. Detection of multiple or even large quantities of uncatalogued objects;
2. Disappearance or significant parametric deviations of catalogued objects.

The critical challenge in rapidly detecting satellite breakup events lies in effectively responding to and processing the emergence of multiple new unidentified objects in space catalogs.

To address this question, we developed an automated detection and analysis framework that leverages the characteristic three-dimensional spatiotemporal density distribution of breakup debris clusters. The method employs the density-based spatial clustering of applications with noise (DBSCAN) algorithm to process unidentified observational targets, automatically identifies potential satellite breakup events upon detecting substantial unidentified objects, and conducts orbital backtracking analysis to verify breakup occurrences and reconstruct preliminary event scenarios.

The specific objectives of this study are:

1. To develop an automated detection method for satellite breakup events using density-based clustering with orbital backtracking analysis;
2. To design a strategy that balances detection sensitivity with noise robustness in operational scenarios;
3. To validate the method across different orbital regimes using real-world breakup events.

Density-based clustering algorithms have previously been applied to space debris analysis. For instance, Wu and Rosengren [14] and Jiang [15] employed DBSCAN to classify debris clouds generated by breakup events, distinguishing fragments originating from different sources. Lorenzo Giudici et al. [16] developed a density-based space debris evolution model, applied to estimate the evolution of the space debris population in low-Earth orbit up to 200 years from the reference epoch. These approaches focus primarily on post-event classification using known debris catalogs. As a comparison, this

study addresses the operational challenge of real-time detection from uncorrelated targets without prior knowledge. This work introduces several key advancements:

1. A two-layer clustering strategy specifically designed for real-time breakup detection in operational SSA scenarios;
2. Temporal density optimization through orbital backtracking to identify the epoch of maximum cluster convergence;
3. Application across both LEO and GEO regimes with parameter adaptation strategies.

Through validation with the historic Cosmos 2251-Iridium 33 collision and the recent Intelsat 33E breakup event, we demonstrate that the proposed method can effectively reconstruct breakup scenarios without prior knowledge, providing timely estimation of breakup time and location for operational space situational awareness applications.

2. Methodology

2.1. DBSCAN Algorithm

In the fields of data analysis and machine learning, DBSCAN (density-based spatial clustering of applications with noise) is a density-based spatial clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu in 1996 [17]. It automatically discovers clusters of arbitrary shapes through the density distribution of data points while effectively handling noise points, making it suitable for complex and noisy datasets.

Within the DBSCAN algorithm, density is defined by the number of points contained within a specified radius. The key idea of the DBSCAN algorithm is that for each point in a cluster, the neighborhood of a given radius must contain at least a minimum number of points, i.e., the density within the neighborhood must exceed a predefined threshold. This leads to the core principles and concepts of DBSCAN:

1. Key parameters
   (a) Neighborhood radius ε: defines the physical scope around a point p, constituting its Eps-neighborhood $N_\varepsilon(p)$:

$$N_\varepsilon(p) = \{q \in D | dist(p,q)| \leq \varepsilon\} \tag{1}$$

   (b) Minimum points $MinPts$: specifies the threshold quantity of points required within the neighborhood to satisfy DBSCAN's density criterion.
2. Core concepts
   (a) Core point: A point p is designated as a core point $|N_\varepsilon(p)| \geq MinPts$;
   (b) Border point: A non-core point situated within the ε-neighborhood of any core point;
   (c) Directly density-reachable: If point p lies within the ε-neighborhood of another core point q, then p is directly density-reachable from q:

$$\begin{cases} p \in N_\varepsilon(q) \\ |N_\varepsilon(q)| \geq MinPts \end{cases} \tag{2}$$

   (d) Density reachable: If point p is not directly density reachable from a core point p_n, but there exists a chain of points $p_1, p_2, ..., p_n$ where each consecutive pair

is directly density reachable, and p is directly density reachable from p_1, then p is density reachable from p_n:

$$\begin{cases} p \in N_\varepsilon(core\ point : p_1) \\ p_1 \in N_\varepsilon(core\ point : p_2) \\ ..... \\ p_{n-1} \in N_\varepsilon(core\ point : p_n) \end{cases} \tag{3}$$

(e) Density connected: If there exists a core point p such that both points o and q are density-reachable from p, then o and q are density connected;

(f) Cluster: The maximal set of density-connected points constitutes a cluster, whose mathematical expression and properties are defined as follows:

$$C = \begin{cases} N_\varepsilon(core\ point : p) \\ p\ density\ reachable\ to\ q\ (Maximality) \end{cases} \tag{4}$$

$$\forall p, q \in C,\ p\ is\ density\ connected\ to\ q \tag{5}$$

The operational steps of the DBSCAN algorithm can be succinctly described as follows:

1. Initialization:
   (a) Traverse all unvisited points in the dataset;
   (b) Select an arbitrary unvisited point p.
2. Core Point Verification:
   (a) Calculate the ε-neighborhood $N_\varepsilon(p)$;
   (b) If points within $N_\varepsilon(p) \geq MinPts$:
      i. Designate p as a core point;
      ii. Connect all points within $N_\varepsilon(p)$ to create new cluster C;
      iii. Expand cluster by merging all density-reachable points.
3. Cluster Expansion:
   (a) Iteratively add all directly density-reachable points to C;
   (b) Include border points within core points' neighborhoods.
4. noise Identification:
   (a) Mark non-clustered points as noise.
5. Termination:
   (a) Repeat until all points are visited and classified.

2.2. A Detection Method for Satellite Breakup Events Based on Two-Layer Space Debris Clustering and Orbital Backtracking

This section presents a detection method designed to rapidly identify debris clusters and determine breakup event parameters in operational space situational awareness scenarios. When a satellite breaks up in orbit—whether due to collision, explosion, or other fragmentation mechanisms—the resulting debris initially share a common spatial origin and subsequently disperse according to orbital dynamics. This creates a transient, high-density cluster of objects in three-dimensional space that can be detected through density-based clustering analysis.

The operational scenario addressed by this method involves the following sequence: space surveillance networks continuously monitor near-Earth space and detect uncorrelated targets whose origins are unknown. When a breakup event occurs, newly generated debris appear as emergent high-density clusters within the population of tracked objects.

The method employs the DBSCAN algorithm to identify these dense clusters from the background population of unrelated objects, then applies orbital backpropagation to estimate the breakup time and location.

Figure 2 illustrates this clustering phenomenon using the Cosmos 2251-Iridium 33 collision as an example. At the breakup epoch, debris from both satellites form a discretely distributed point cluster with significant spatial aggregation near the collision origin, while independently orbiting satellites or debris appear as noise points external to the cluster. This real-world distribution—with dense signal clusters embedded in a dispersed noise background—is well-suited for analysis using the DBSCAN algorithm, which robustly handles datasets containing noise.

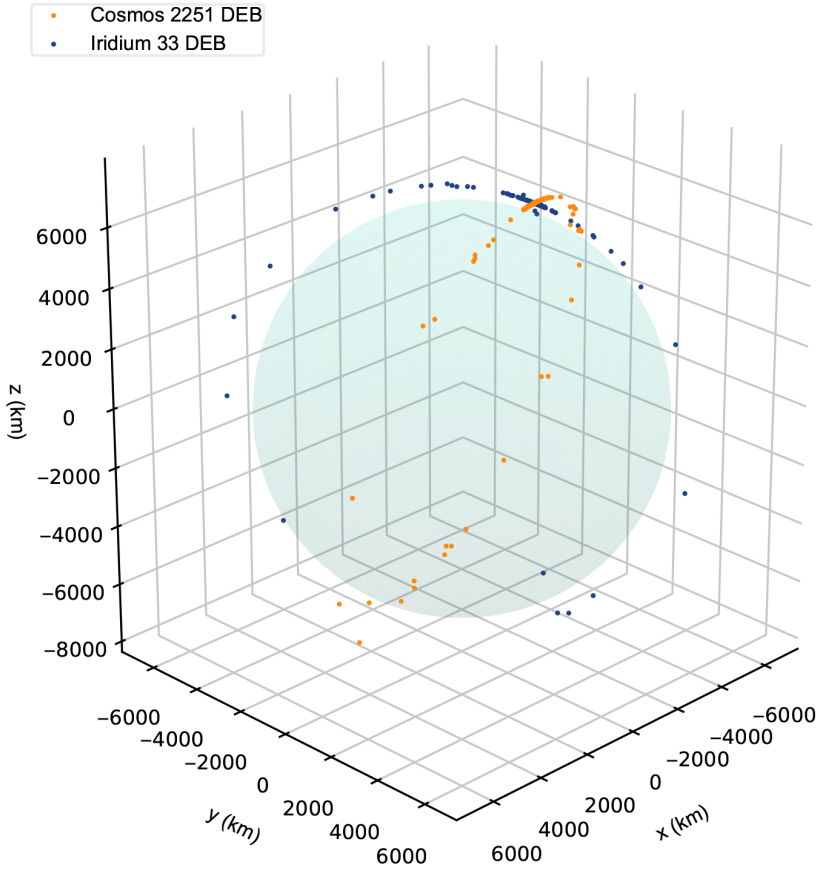


Figure 2. Space debris of Cosmos 2251 and Iridium 33 satellites at collision time 10 February 2009 16:55:59 UTC. (The green shading represents a spherical Earth (radius = 6378.135 km).)

Our objective is to employ the DBSCAN algorithm to identify clusters within this point set that correspond to the collision breakup event between the Cosmos 2251 and Iridium 33 satellites, backtrack the orbital trajectories of the debris clusters, and thereby determine the breakup origin and parent objects to analyze the causes of the event. The designed detection and analysis procedures for breakup events are structured as follows (as illustrated in Figure 3):

1. Begin: Input the target orbital elements to be clustered;
2. Input parameter: Eps-neighborhood radius 1 ε_1, Eps-neighborhood radius 2 ε_2, Minimum number of points $MinPts$;
3. Calculation of t_0: Set the initial time t_0 by taking the earliest orbit epoch of the target set as time t_0.
4. Calculation of target positions at t_0: Compute the positional coordinates of each object at the reference time t_0.

5. First-layer clustering: At time t_0, initiate the first-layer clustering by selecting a larger neighborhood radius ε_1. This step serves two primary objectives: (a) elimination of outliers; (b) preliminary clustering to identify potential fragment clusters resulting from breakup events. The operational scenario assumes that terminal nodes in situational awareness networks detect debris clouds generated by breakup events. Since these debris clouds might not yet have reached their maximum density at the origin of breakup, the selection of an extended neighborhood radius becomes necessary for effective detection.
6. Second-layer clustering: For each cluster set $\{A_0, B_0, C_0, ...\}$ generated by the first-layer clustering, conduct trajectory backtracking and perform second-layer clustering with dual objectives: (a) temporal density optimization through trajectory backtracking of primary clusters to determine the timepoint of maximum cluster density for secondary clustering; (b) exclusion of extraneous objects incorporated during the first-layer clustering. The implementation workflow for individual clusters proceeds as follows:
    a Define the time interval for trajectory backtracking: The time interval for trajectory backtracking is defined as $(t_{e0} - \Delta d, t_{e1})$, where t_{e0} corresponds to the earliest orbital epoch (in Julian Date, JD) within the cluster, and t_{e1} denotes the latest orbital epoch in the cluster. The value of the interval Δd (in days) depends on the coverage capability and efficiency of situational awareness.
    b Determine the moment t_c when the average distance within the target set reaches its minimum during the specified timeframe, which will serve as the initiation time for the secondary clustering process.
    c Second-layer clustering: By selecting a smaller neighborhood radius ε_2, the second-layer clustering is initiated at time t_c, yielding refined clusters after the removal of outliers.
7. Satellite breakup event elements calculation: For clusters $\{A, B, C, ...\}$ obtained after the second-layer clustering, perform steps 6(a-b). The objective of this phase is to determine the temporal occurrence and spatial coordinates of the satellite breakup event associated with each cluster. The critical moment t_c when the average inter-object distance within the cluster reaches its minimum is identified as the breakup timestamp, while the cluster centroid is established as the spatial origin of the breakup event.

Step 6.b. We mention a concept: average distance. The average distance within a cluster is used to characterize the spatial density of debris objects. For a cluster C containing n objects with positions $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in the GCRS Cartesian coordinate system, the average distance d_{avg} is computed as follows:

$$d_{\text{avg}}(C) = \frac{1}{C(n,2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\| = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\| \tag{6}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the Euclidean distance between objects i and j, and $C(n,2) = n(n-1)/2$ is the number of unique object pairs in the cluster. This metric represents the mean pairwise distance among all objects within the cluster and serves as an indicator of cluster compactness: a smaller d_{avg} value indicates that the objects are more densely concentrated, while a larger value suggests greater spatial dispersion.

For clusters containing only a single object ($n = 1$), the average distance is undefined and such clusters are excluded from the convergence analysis. The breakup epoch t_c is identified as the time at which d_{avg} achieves its minimum value over the search interval, as the fragments are initially co-located and subsequently disperse due to differential orbital dynamics.

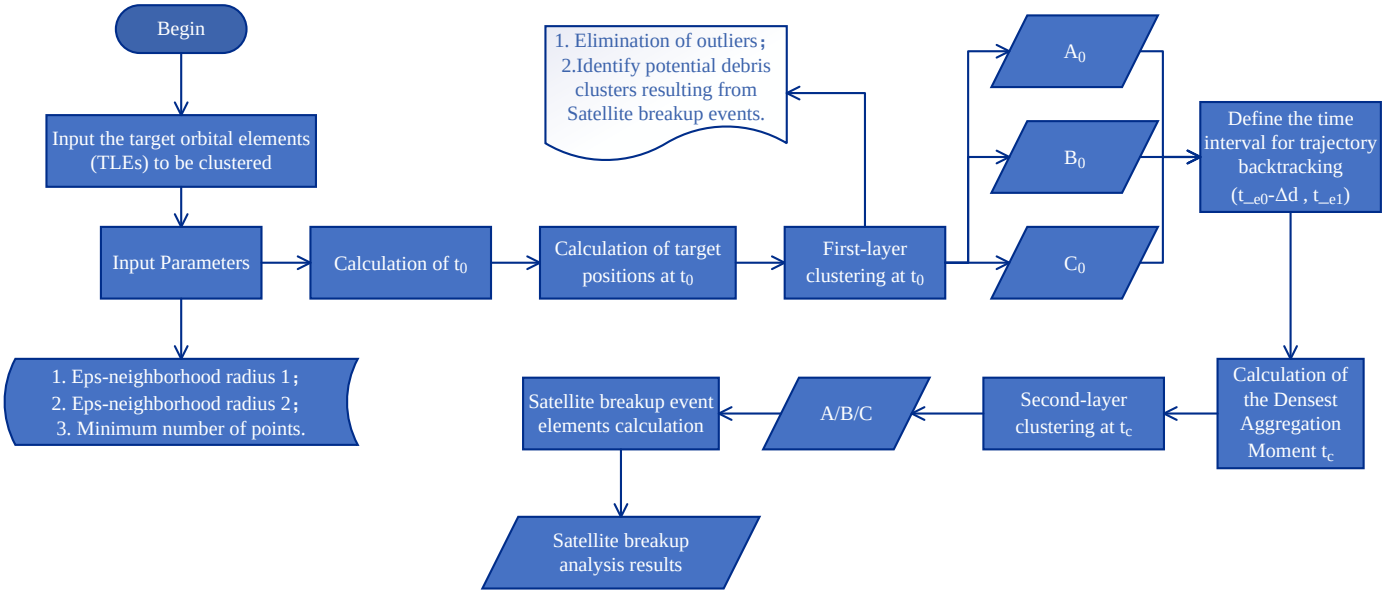


Figure 3. Method for detecting satellite breakup events.

2.3. Orbital Propagation Model

The orbital state propagation in this study employs the SGP4/SDP4 model [18], which is the standard propagator consistent with the two-line element (TLE) data format. The SGP4 (Simplified General Perturbations 4) model is applied to near-Earth objects with orbital periods of less than 225 min, while the SDP4 (Simplified Deep-space Perturbations 4) model is applied to deep-space objects including geosynchronous satellites.

The SGP4/SDP4 algorithm uses the mean orbital elements contained in TLEs as input, including the inclination, right ascension of ascending node, eccentricity, argument of perigee, mean anomaly, mean motion, and the drag term B*. The propagation procedure involves converting these mean elements to osculating positions and velocities at specified epochs using analytical expressions that account for secular and periodic perturbations from Earth's non-spherical gravitational field, atmospheric drag, and for deep-space objects, third-body gravitational effects from the Sun and Moon as well as solar radiation pressure.

The choice of SGP4/SDP4 is appropriate for this study for the following reasons. First, it is the standard propagator used by Space-Track.org for generating TLEs, ensuring compatibility with the input data format. Second, in our operational scenario, we lack prior information about breakup events and do not have accurate area-to-mass ratios or other physical parameters for the detected objects. Under these conditions, using a simplified general perturbations model is reasonable. Third, the intended application scenario of our method assumes detection of breakup events as rapidly as possible—ideally within hours to approximately 24 h after the event occurs, when surveillance networks first observe large numbers of uncorrelated targets. Over such short timescales, it is not possible to derive precise atmospheric drag modeling based on long-term orbital evolution, making the simplified perturbation approach appropriate. Fourth, the model is computationally efficient for processing large catalogs of objects in operational scenarios where near-real-time analysis is required.

Regarding the model's suitability, SGP4/SDP4 is well-suited for our application scenario. As a widely adopted standard in both academic research and operational space surveillance systems, it provides an appropriate balance between computational efficiency and propagation accuracy for the intended use case. The model's simplified perturbation approach is consistent with the nature of the input data (TLEs) and the operational requirements (rapid processing of large object catalogs). Since our method is designed

to detect breakup events within hours to approximately 24 h after occurrence, the short propagation timescales align well with SGP4/SDP4's capabilities. The primary sources of localization error in our analysis stem from TLE measurement uncertainties and the inherent Delta-V distribution of fragments. Like any propagation model, SGP4/SDP4 has inherent assumptions and approximations, but these do not impede its suitability for the operational scenario addressed in this work.

2.4. SGP4 Propagation and Coordinate Transformation Implementation

This section provides the technical details necessary for reproducing the orbital propagation and coordinate transformations reported in this study.

The orbital propagation and coordinate transformation pipeline is implemented using Python 3.10 and a Python library: Skyfield (Version 1.53) (https://ui.adsabs.harvard.edu/abs/2019ascl.soft07024R/abstract, accessed on 12 April 2026), which performs coordinate transformations in accordance with the IERS Conventions 2010 [19] using the IAU-2006/2000A precession-nutation model.

In our pipeline, the propagation is performed as follows:

1. Load TLE data using `skyfield.sgp4lib.EarthSatellite.from_tle_lines()`.
2. Propagate to target epoch using `satellite.at(time)`, which returns a `Geocentric` position object.
3. Extract the GCRS (Geocentric Celestial Reference System) position vector via `position.xyz.km`.
4. Convert to ITRS (International Terrestrial Reference System, Earth-fixed frame) using `position.itrs_xyz.xyz.km`.
5. Obtain geodetic coordinates (latitude, longitude, altitude) based on the WGS-84 ellipsoid model using `satellite.subpoint()`.

The GCRS is a coordinate system within the ICRF (International Celestial Reference Frame) framework, with its origin at the Earth's center of mass. The transformation from GCRS to ITRS accounts for precession, nutation, polar motion (typically $<$0.5 arcseconds, corresponding to $\sim$15 m at the Earth's surface), and UT1-UTC corrections (typically $<$1 s, acquired from IERS Bulletin A). The geodetic coordinates are derived from ITRS Cartesian coordinates using the WGS-84 ellipsoid following the iterative algorithm of Bowring [20]. The complete transformation chain can be summarized as:

$$\text{TLE} \xrightarrow{\text{SGP4}} \text{TEME} \xrightarrow{\text{IAU-2006/2000A}} \text{GCRS} \xrightarrow{\text{GAST+PM}} \text{ITRS} \xrightarrow{\text{WGS-84}} (\phi, \lambda, h) \tag{7}$$

3. Results

This study employs two real-world cases to validate the effectiveness of the satellite breakup detection methodology: the collision-induced breakup event between Cosmos 2251 and Iridium 33 satellites, and the recent in-orbit breakup of the geostationary communication satellite Intelsat 33E.

3.1. Analysis of Collision-Induced Breakup Event Between Cosmos 2251 and Iridium 33 Satellites

According to T.S. Kelso's analysis of the 2009 collision breakup event between Cosmos 2251 and Iridium 33 satellites [21], the collision warning network issued an alert on 10 February 2009, predicting a close approach of 584 m between Iridium 33 and Cosmos 2251 at 16:55:59.806 UTC. At the forecast conjunction time, Iridium 33 suddenly went silent, and then the US Space Surveillance Network (SSN) subsequently reported that they were tracking debris clouds in both the Iridium 33 and Cosmos 2251 orbits, confirming a collision. The central positions of both satellites at the collision time in the GCRS coordinate system are $p_{collision} = (-1465.144,\ 1603.321,\ 6808.831)$.

The breakup event detection algorithm we designed aims to rapidly analyze the emergence of multiple uncatalogued objects and determine whether a breakup event has occurred, typically within a short timeframe (1–2 days). Regarding the Cosmos 2251 and Iridium 33 collision event, the U.S. Air Force Space Command (now the U.S. Space Force) first released orbital data for partial debris on 17 February 2009, seven days post-collision. Additional debris continued to be discovered and cataloged in subsequent years. For debris cataloged long after the collision event, whose initial orbital epochs substantially deviate from the breakup date, orbital backtracking would incur significant propagation errors. Therefore, our analysis exclusively incorporates debris detected within 90 days post-collision, comprising 604 debris from Cosmos 2251 and 278 debris from Iridium 33. Note that the 90-day temporal limit was empirically determined through tests with progressively older debris data, beyond which cluster convergence became unreliable. The spatial distribution of their initial orbital positions at the breakup epoch is illustrated in Figure 2.

Furthermore, to analyze this breakup event utilizing orbital elements with epochs exceeding one week post-breakup, and to validate the breakup event detection methodology described in Section 2 (hereinafter referred to as "the Method"), we must establish a simulated scenario with specified constraints on the Method's input parameters, as detailed below.

The simulated scenario assumes the space situational awareness network detected debris two hours post-breakup, with the observation timestamp recorded as: t_{obs} = 10 February 2009 18:55:59.806 UTC. At this time, the collision-generated debris and 100 simulated noise (randomly selected non-breakup event-related objects as uncatalogued objects near both satellites' orbits, with orbital data sourced from Space-Track.org) formed the spatial distribution shown in Figure 4. Figure 4 reveals two distinct high-density clusters corresponding to the majority of debris from Cosmos 2251 and Iridium 33 satellites. Our objective is to employ the DBSCAN algorithm to identify breakup debris clusters within the uncatalogued object population (comprising both debris and simulated noise), followed by orbital backtracking to determine the breakup's temporal and geospatial parameters.

Additionally, we impose the following specific constraints on the methodological inputs:

1. In the method, the first-layer clustering t_0 is defined as $t_0 = t_{obs}$. The conventional procedure requires capturing debris clusters through observations and determining the initial data point by their initial orbital epoch, which is then designated as the start time of the first-layer clustering. However, for this satellite breakup event, we only have the first available orbital data of breakup debris published on Space-Track.org, whose epoch occurs one week after the satellites disintegrate in orbit. Therefore, we anchor t_0 to the simulated scenario time t_{obs}.
2. Prior to the second-layer clustering, the start time of orbit backtracking t_{e0} is similarly designated as t_{obs}, following the same rationale. The end time t_{e1} should also not be set as the latest epoch in the predetermined target set of the method, as the target set contains orbital data three months after the satellite breakup event. Using these epochs would result in excessively long orbit backtracking intervals and increased computational complexity. In this scenario, t_{e1} can be specified as the latest epoch in the noise target set: t_{e1} = 10 February 2009 23:00:13 UTC.
3. Within the orbit backtracking interval, $\Delta d = 1$, simulating an observation gap window of 1 day.

Table 1 presents the orbit elements at the observation epoch t_{obs} for 10 representative debris from the Cosmos 2251 and Iridium 33 satellite breakup events, along with 10 noise at their respective orbital epochs, used in our computations.

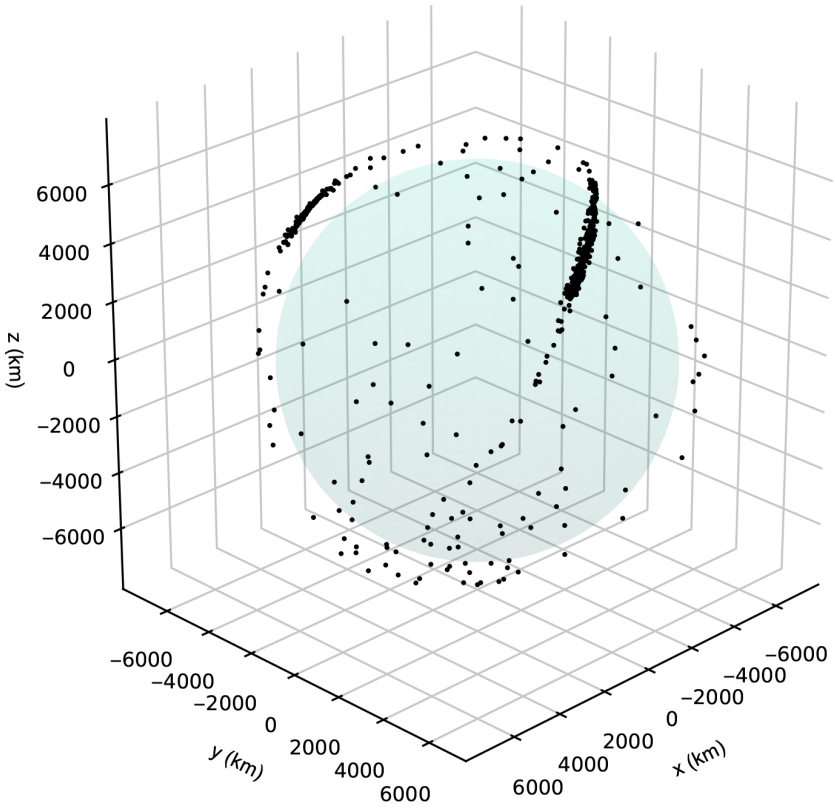


Figure 4. Positions of debris from satellite Cosmos 2251 and satellite Iridium 33 with other objects at 10 February 2009 18:55:59 UTC (2 h after breakup time). (The green shading represents a spherical Earth (radius = 6378.135 km).)

Under specified constraints, the input parameters for this analysis are: neighborhood radius $\varepsilon_1 = 2000$ km, $\varepsilon_2 = 500$ km, and $MinPts = 5$. After two-layer clustering processing, the method identified three clusters in the input uncorrelated target set, as shown in Figure 5a: Cluster C_1 (marked as Iridium 33 debris cluster) contains 252 targets, all of which are Iridium 33 debris. For the input Iridium 33 debris samples, Cluster C_1 missed 24 debris, with two misclassified into Cluster C_3, yielding a recall rate of 90.65%, demonstrating effective performance. Cluster C_2 (marked as Cosmos 2251 debris cluster) comprises 569 targets, exclusively Cosmos 2251 debris, with 34 missed detections and a sample recall rate of 94.37%. Cluster C_3, designated as the noise cluster, contains five targets (three input noise points and two Iridium 33 debris) that formed a small isolated cluster due to their spatial distribution within the neighborhood radius used for clustering and subsequent orbital backtracking convergence.

Comparative analysis between Clusters C_1–C_2 and original samples reveals that most missed targets were independent debris spatially separated from the debris clusters at t_0. This phenomenon may stem from potential orbital errors in publicly available TLE data. The 90-day post-breakup orbital data used in this study introduces additional uncertainties, as long-term propagation amplifies inaccuracies in B* Drag estimation for two-line elements (TLEs) during orbit backtracking. Through two-layer clustering and backtracking phases, the method identified cluster convergence moments (minimum intra-cluster distance epochs) and spatial centroids, as Table 2.

Table 1. Orbit elements of examples of debris and noise.

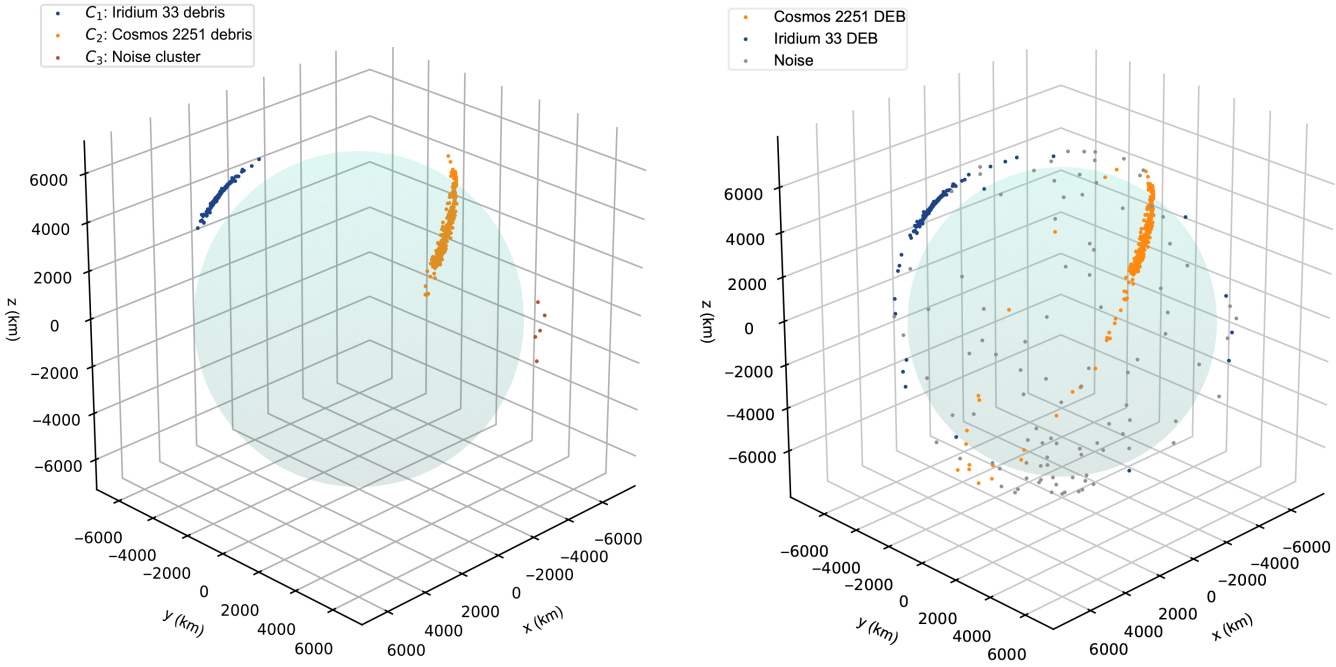
Norad ID	Epoch (UTC)	Semi-Major Axis (km)	Inclination (deg)	Eccentricity
Cosmos 2251 Debris				
22675	2009/02/10 18:09:29.10	7172.234	74.059	0.003
33757	2009/02/17 07:48:40.17	7170.824	74.051	0.003
33758	2009/02/17 14:32:34.17	7171.483	74.052	0.003
33759	2009/02/17 14:22:48.17	7166.781	74.043	0.002
33760	2009/02/17 14:19:53.17	7165.403	74.047	0.002
33761	2009/02/17 14:24:24.17	7167.578	74.053	0.003
33762	2009/02/17 14:25:10.17	7167.936	74.057	0.002
33763	2009/02/17 14:43:15.17	7176.575	73.970	0.003
33764	2009/02/17 07:45:06.17	7169.032	74.052	0.003
33765	2009/02/17 08:18:40.17	7135.897	74.043	0.020
Iridium 33 Debris				
24946	2009/02/10 18:16:11.10	7161.966	86.447	0.001
33771	2009/02/17 14:20:17.17	7161.235	86.458	0.001
33772	2009/02/17 14:11:17.17	7205.422	86.484	0.007
33773	2009/02/17 14:18:15.17	7160.209	86.453	0.002
33774	2009/02/17 10:06:35.17	7184.752	86.468	0.004
33775	2009/02/17 14:45:01.17	7173.098	86.417	0.003
33776	2009/02/17 14:44:35.17	7172.854	86.454	0.003
33777	2009/02/17 14:20:04.17	7161.084	86.437	0.001
33778	2009/02/17 10:08:10.17	7185.496	86.366	0.004
33849	2009/02/19 15:13:28.19	7244.344	86.182	0.013
Noise				
27844	2009/02/10 00:07:32.10	7207.509	98.751	0.002
18585	2009/02/10 00:12:38.10	7166.354	73.998	0.001
27846	2009/02/10 00:13:59.10	7206.347	98.745	0.002
19468	2009/02/10 00:17:47.10	7266.704	98.772	0.006
3229	2009/02/10 00:19:18.10	7034.939	74.027	0.005
7364	2009/02/10 00:21:20.10	7258.643	81.200	0.006
25078	2009/02/10 00:22:11.10	7154.289	86.370	0.001
27847	2009/02/10 00:24:08.10	7206.403	98.745	0.002
14373	2009/02/10 00:25:26.10	7007.035	82.518	0.002
28499	2009/02/10 00:25:37.10	7023.010	98.252	0.009

Table 2. Cosmos 2251—Iridium 33 breakup event analysis result.

Clusters	Time UTC	Center Position (LLA)	Center Position (GCRS)
C_1: Iridium 33 debris	2009/02/10 16:55:55	(97.694° E, 72.242° N, 787.698 km)	(−1473.044, 1617.071, 6803.456)
C_2: Cosmos 2251 debris	2009/02/10 16:55:55	(97.057° E, 72.598° N, 760.475 km)	(−1421.177, 1595.636, 6790.916)

Both satellite debris clusters converged at 10 February 2009 16:55:55 UTC. Figure 6 illustrates the spatial distribution of the clusters at their convergence moment. This time moment deviates by 5 s from the actual collision time when Cosmos 2251 and Iridium 33 satellites are encountered. By averaging the central positions of both clusters as the analyzed breakup position (GCRS): $p_{cluster} = (-1447.114,\ 1606.353,\ 6797.186)$, a 21.68 km separation is observed from the predicted collision position $p_{collision}$ of the two satellites.

The discrepancies in cluster convergence time and breakup position primarily originate from orbital inaccuracies in the debris dataset and aged orbital epochs, which have resulted in a pre-existing spatial distribution of debris at the breakup moment, preventing complete backtracking to the predicted intersection point of the Cosmos 2251 and Iridium 33 satellites' pre-collision trajectories. In summary, this method demonstrates robust applicability to the collision-induced breakup event of Cosmos 2251 and Iridium 33 satellites. Comprehensive sample analysis shows that the method achieves 98.92% accuracy and 93.20% recall rate for debris samples (positive samples) from both satellites.



(**a**) (**b**)

Figure 5. Comparison between clusters generated by methods and samples. (**a**) Debris clusters made by the method at 10 February 2009 18:55:59 UTC. (**b**) All samples' positions at 10 February 2009 18:55:59 UTC.

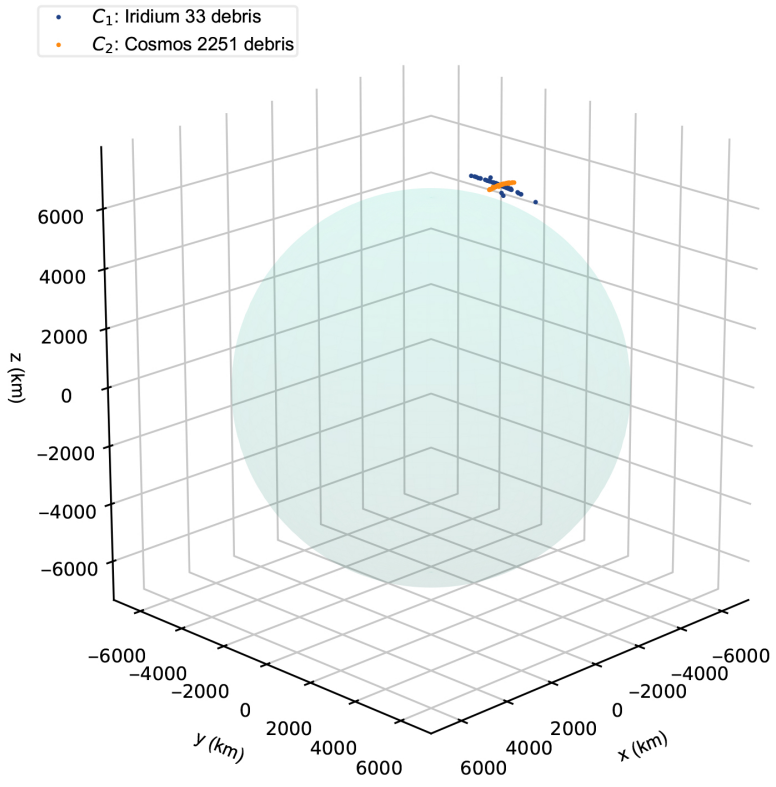


Figure 6. Debris clusters of Cosmos 2251 satellite and Iridium 33 satellite at clusters' gathering time 10 February 2009 16:55:55 UTC.

https://doi.org/10.3390/aerospace13050396

3.2. Sensitivity Analysis of DBSCAN Parameters

The MinPts parameter is set to five based on the characteristic fragmentation patterns of satellite breakup events. Satellite fragmentation, whether caused by hypervelocity collision, explosive failure, or deliberate destruction, typically generates multiple debris objects. Historical data from past breakup events indicates that even low-energy fragmentation incidents produce at least three cataloged fragments, while high-energy collisions can generate hundreds to thousands of trackable debris. The MinPts value of five thus serves as a conservative threshold that effectively distinguishes between true breakup-related clusters and random orbital coincidences, preventing background objects from being misclassified as debris.

To systematically evaluate the impact of parameter selection on detection performance, we conducted a comprehensive sensitivity analysis for the Cosmos 2251–Iridium 33 case. Specifically, we systematically varied the two distance thresholds governing the two-layer clustering procedure: ε_1 (the neighborhood radius in orbital element space for Layer 1) and ε_2 (the neighborhood radius in retro-propagated Cartesian space for Layer 2). The impact of these parameter variations on detection performance is summarized in Table 3.

Table 3. Sensitivity analysis of DBSCAN parameters on clustering performance (100 noise).

ε_1 (km)	ε_2 (km)	TP	FP	TN	FN	Precision	Accuracy	Recall Rate	F1 Score
500	50	773	0	100	109	100.00%	88.90%	87.64%	0.93
500	100	783	0	100	99	100.00%	89.92%	88.78%	0.94
500	200	812	0	100	70	100.00%	92.87%	92.06%	0.96
500	500	819	0	100	63	100.00%	93.58%	92.86%	0.96
1000	50	671	0	100	211	100.00%	78.51%	76.08%	0.86
1000	100	766	0	100	116	100.00%	88.19%	86.85%	0.93
1000	200	809	0	100	73	100.00%	92.57%	91.72%	0.96
1000	500	822	1	99	60	99.88%	93.79%	93.20%	0.96
2000	50	597	0	100	285	100.00%	70.98%	67.69%	0.81
2000	100	753	0	100	129	100.00%	86.86%	85.37%	0.92
2000	200	808	0	100	74	100.00%	92.46%	91.61%	0.96
2000	500	822	0	100	60	100.00%	93.89%	93.20%	0.96

The parameter selection ($\varepsilon_1 = 2000$ km, $\varepsilon_2 = 500$ km, MinPts = 5) was based on maximizing both clustering accuracy and recall while maintaining physically meaningful cluster separation. The ε_1 value corresponds approximately to the maximum expected dispersion of fragments within the first few hours post-breakup for LEO collisions, while ε_2 represents the characteristic scale of fragment clouds at convergence.

As can be seen from Table 3, the Cosmos 2251-Iridium 33 collision event produced a substantial number of debris fragments with high spatial density, facilitating the propagation of the DBSCAN algorithm and enabling successful clustering into debris clusters. When the secondary threshold parameter ε_2 exceeds 50 km, the recall rate of the algorithm remains consistent across different values of the primary threshold ε_1, with F1-scores stabilizing in the range of 0.92–0.96. This indicates that during the first-layer clustering, the algorithm successfully identifies and clusters the majority of debris under these three threshold settings. In this LEO collision breakup case, the secondary parameter ε_2 plays a more significant role, with a larger ε_2 range helping the algorithm improve both recall rate and precision, while maintaining high F1-scores that balance precision and recall.

We further evaluated cluster-level performance indicators to assess the quality of identified debris groups. A confusion-matrix-style summary for this configuration is presented in Figure 7. The two-layer DBSCAN framework successfully identified both debris clusters (Cosmos 2251 and Iridium 33) as distinct, spatially coherent groups. The cluster purity—defined as the proportion of true debris within each detected cluster—remained at

100% for the optimal parameter setting ($\varepsilon_1 = 2000$ km, $\varepsilon_2 = 500$ km), indicating that no background objects were erroneously incorporated into the debris clusters.

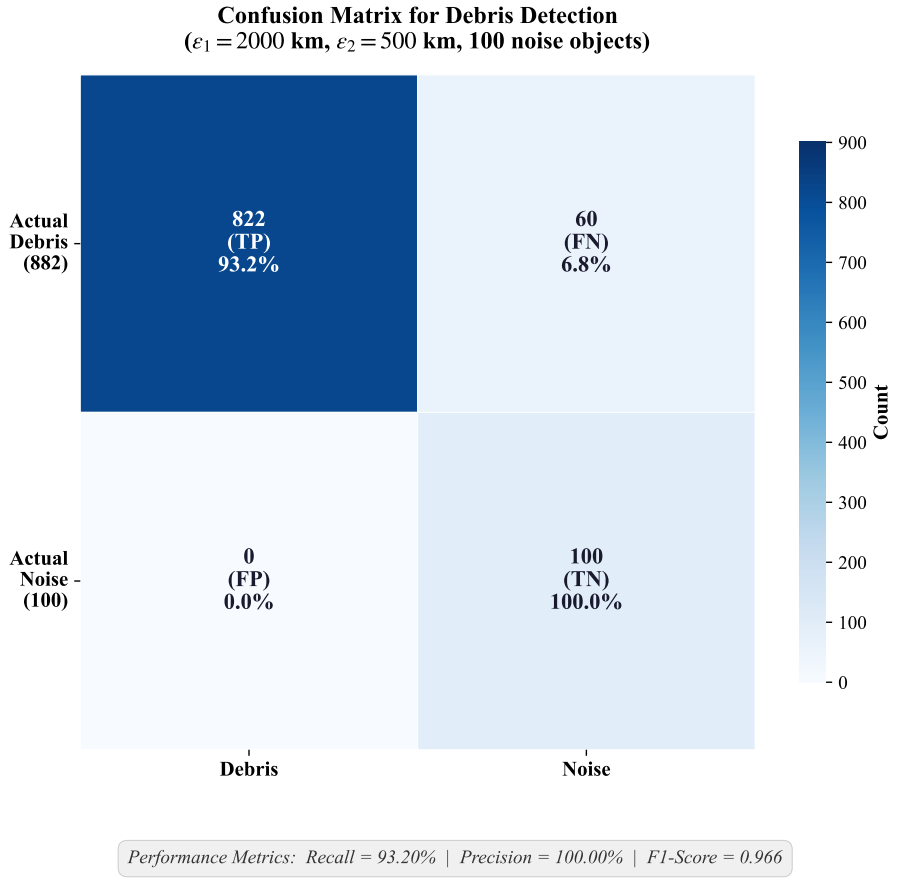


Figure 7. Confusion matrix for debris detection using optimal parameters ($\varepsilon_1 = 2000$ km, $\varepsilon_2 = 500$ km, 100 noise objects).

The confusion matrix reveals that the method correctly identified 822 out of 882 debris objects (93.20% recall) while maintaining zero false positives, demonstrating robust separation between breakup debris and the background population. The 60 false negatives correspond to debris that were spatially separated from the main clusters at the observation epoch, likely due to orbital dispersion or TLE measurement uncertainties.

3.3. Sensitivity and Robustness of Clustering Performance

To validate the method's robustness in noisy operational environments, we augmented the Cosmos 2251-Iridium 33 debris set with the entire population of other objects in LEO. Figure 8 shows their positions at t_{obs}, illustrating a realistic scenario where breakup debris must be distinguished from all other background traffic.

Under high-noise conditions, the breakup event detection method based on two-layer space debris clustering and orbital backtracking continues to demonstrate robust performance. As shown in Table 4, with parameters set to $\varepsilon_1 = 1000$ km, $\varepsilon_2 = 200$ km, and $MinPts = 5$, the method achieves an accuracy of 98.05% and a recall rate of 92.86%. Unlike normal-noise environments, under high-noise conditions, the distance parameters of the DBSCAN algorithm should not be set too large; otherwise, debris clusters are prone to incorporating more noise, leading to a rapid decline in clustering accuracy despite favorable recall rates. This scenario underscores the importance of the two-layer clustering approach: Figure 9 demonstrates that after the second-layer clustering, the algorithm

effectively eliminates most of the noise, enabling a more accurate reconstruction of the breakup scenario and yielding results closer to the actual event time and location.

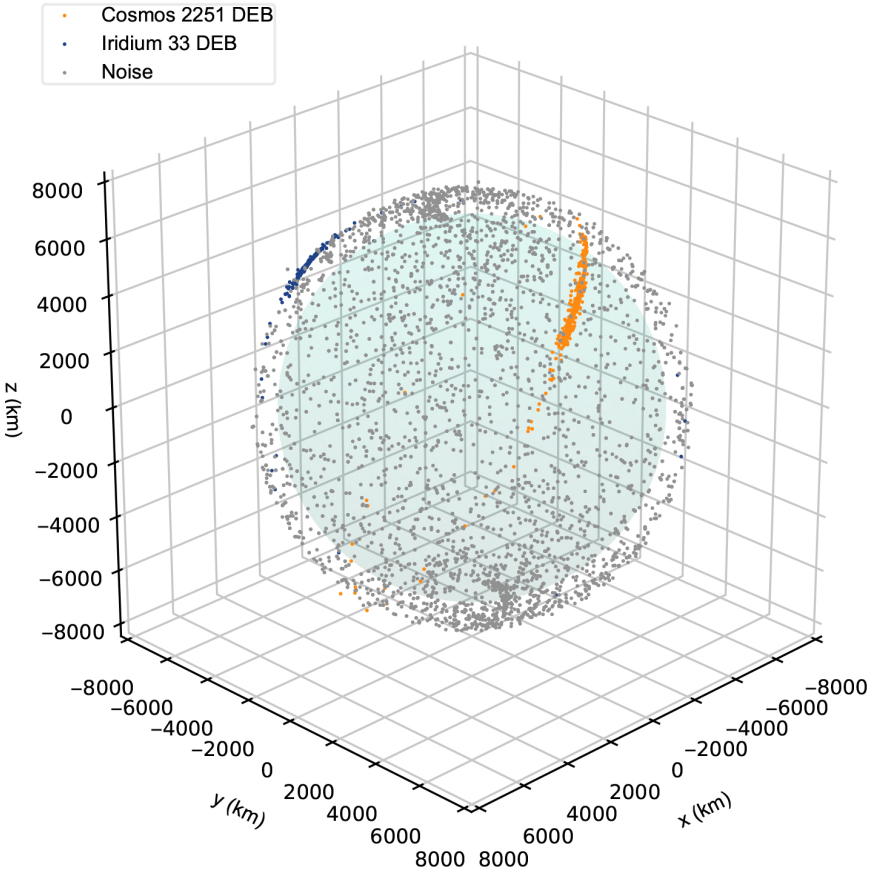


Figure 8. Positions of debris from satellite Cosmos 2251 and satellite Iridium 33 with the entire population of other objects in LEO at 10 February 2009 18:55:59 UTC.

Table 4. Parameter analysis under high-noise conditions.

ε_1 (km)	ε_2 (km)	TP	FP	TN	FN	Precision	Accuracy	Recall Rate	F1 Score
500	50	582	3	4190	300	99.49%	94.03%	65.99%	0.79
500	100	748	5	4188	134	99.34%	97.26%	84.81%	0.91
500	200	810	10	4183	72	98.78%	98.38%	91.84%	0.95
500	500	836	46	3522	671	94.78%	85.87%	55.47%	0.70
1000	50	775	2	4191	107	99.74%	97.85%	87.87%	0.93
1000	100	819	18	4175	63	97.85%	98.40%	92.86%	0.95
1000	200	819	18	4175	63	97.85%	99.00%	92.86%	0.95
1000	500	838	44	3224	969	95.01%	80.04%	46.38%	0.62
2000	50	775	2	4191	107	99.74%	97.85%	87.87%	0.93
2000	100	785	3	4190	97	99.62%	98.03%	89.00%	0.94
2000	200	818	17	4176	64	97.96%	98.40%	92.74%	0.95
2000	500	844	1034	3159	38	44.94%	78.88%	95.69%	0.61

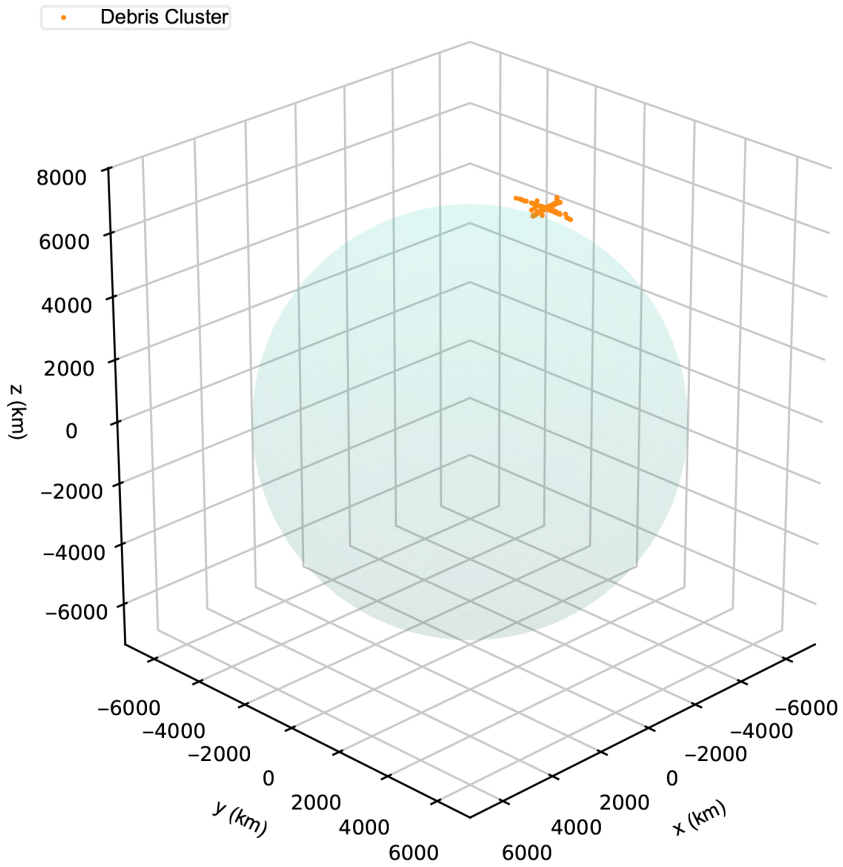


Figure 9. Positions of a single debris cluster from Cosmos 2251 and Iridium 33 satellites identified under high-noise conditions through two-layer clustering ($\varepsilon_1 = 1000$ km, $\varepsilon_2 = 200$ km), shown at the moment of maximum density.

3.4. Robustness Analysis of Breakup Epoch and Fragmentation Locations

The accuracy of estimated breakup time and location depends not only on the correctness of debris cluster identification but also on the robustness of the convergence analysis to variations in input data and algorithm parameters. To quantify this robustness, we conducted a systematic sensitivity analysis using the Cosmos 2251–Iridium 33 collision as a benchmark case, for which authoritative reference values are available from the detailed post-event analyses cited in Section 3.1. The reference breakup time is 10 February 2009 16:55:59.806 UTC, and the reference GCRS position is (−1465.144, 1603.321, 6808.831) km.

The sensitivity analysis varied two key factors: the clustering thresholds (ε_1, ε_2) and the background noise level (100 noise and 4193 noise). For each parameter combination, we computed the deviation of the estimated breakup time (Δt) and position (Δr) from the reference values. When two debris clusters were detected (corresponding to the separate Cosmos 2251 and Iridium 33 debris clouds), the estimated breakup parameters were computed as the average of the two cluster centers. When only one cluster was detected (indicating that the two debris clouds were merged), the parameters were derived directly from that single cluster. Table 5 presents the results for all tested parameter combinations.

In the low-noise scenario (100 background objects), the method consistently identified two distinct debris clusters corresponding to the Cosmos 2251 and Iridium 33 fragments across all parameter combinations. The estimated breakup time, computed as the average of the two cluster convergence times, exhibits strong stability with deviations ranging from −0.21 to +0.29 min (±17 s)—essentially constant across all tested parameters. The position deviation, computed from the average of the two cluster centers, ranges from 46 to 218 km,

with smaller ε_2 values yielding better accuracy. The optimal result (46 km) is achieved with $\varepsilon_1 = 500$ km and $\varepsilon_2 = 200$ km, demonstrating that the averaged cluster centers accurately reconstruct the collision location even when individual clusters may be slightly offset.

Table 5. Sensitivity of estimated breakup parameters to clustering thresholds and noise levels for the Cosmos 2251–Iridium 33 collision.

ε_1 (km)	ε_2 (km)	Noise	Δt (min)	Δr (km)
500	50	100	−0.21	56
500	200	100	−0.21	46
500	500	100	+0.29	211
1000	50	100	+0.29	204
1000	200	100	+0.29	218
1000	500	100	+0.29	204
2000	50	100	+0.29	203
2000	200	100	+0.29	218
2000	500	100	+0.29	208
500	50	4000	+1.21	682
500	200	4000	+7.96	3568
500	500	4000	+67.02	4089
1000	50	4000	−0.48	149
1000	200	4000	−0.48	133
1000	500	4000	−0.48	1535
2000	50	4000	−0.21	69
2000	200	4000	−0.21	54
2000	500	4000	−0.21	1638

Under high-noise conditions (4000+ background objects), the clustering behavior diverges depending on ε_1. For $\varepsilon_1 = 500$ km, the algorithm continues to identify two separate clusters, but cluster contamination by background objects severely degrades the parameter estimates. As shown in Table 5, the time deviation increases to +1.21 to +67 min and position deviation reaches 682–4089 km, with larger ε_2 values exacerbating the degradation. For $\varepsilon_1 \geq 1000$ km, however, the algorithm identifies a single merged cluster containing debris from both satellites. This merging behavior, while reducing the spatial resolution of individual debris cloud characterization, substantially improves the robustness of breakup parameter estimation. For $\varepsilon_1 = 1000$ km with $\varepsilon_2 \leq 200$ km, the time deviation remains at −0.48 min and position deviation at 133–149 km—comparable to the low-noise performance. The optimal high-noise result is achieved with $\varepsilon_1 = 2000$ km and $\varepsilon_2 = 200$ km, yielding a time deviation of −0.21 min and position deviation of only 54 km.

The contrasting behaviors at different ε_1 values reveal an important trade-off. Smaller ε_1 preserves the separation of the two debris clouds, which is desirable for debris cloud characterization but renders the method vulnerable to cluster contamination in high-noise environments. Larger ε_1 causes the two debris clouds to merge into a single cluster, which sacrifices the ability to distinguish individual debris sources but dramatically improves robustness by averaging out contaminating noise objects. This merged-cluster approach leverages the physical fact that the two satellites collided at the same point in space–time, so a combined cluster still converges to the correct breakup parameters as long as the contamination level is not excessive.

The position estimation shows sensitivity to ε_2 even within the merged-cluster regime. For $\varepsilon_1 = 2000$ km, increasing ε_2 from 50 to 200 km actually improves the position accuracy (69 to 54 km), likely because a moderately larger ε_2 captures more true debris objects that improve the convergence statistics. However, excessive $\varepsilon_2 = 500$ km degrades the position

deviation to 1638 km, indicating that permissive thresholds allow substantial background contamination even when the clusters are merged.

These findings have direct implications for parameter selection in operational applications. For LEO collision events in cluttered orbital environments, we recommend $\varepsilon_1 \geq 1000$ km and $\varepsilon_2 \leq 200$ km to ensure robust breakup parameter estimation. Under these conditions, the method provides time estimates accurate to within 1 min and location estimates accurate to within 150 km—sufficient for event confirmation, debris cloud tracking initiation, and conjunction assessment.

3.5. Computational Performance

To evaluate the computational efficiency and scalability of the proposed framework for operational applications, we conducted runtime benchmarks using the Cosmos 2251-Iridium 33 collision dataset with varying noise levels. The experiments were performed on a workstation equipped with an AMD Ryzen 9 9900X processor (12 cores, 24 logical processors) and 32 GB RAM, running the Python 3.10 implementation with the Skyfield library for orbital propagation.

For the scenario with 1000 objects (882 debris fragments and 118 background objects) and a 2-day orbital backtracking window sampled at 1684 time steps, the complete analysis pipeline required approximately 69 s of wall-clock time with peak memory usage of 150 MB. For the high-noise scenario with 5000 objects (882 debris fragments and 4118 background objects) under identical orbital backtracking parameters, the analysis took 7 min 20 s (440 s) with peak memory usage of 350 MB.

The performance exhibits near-linear scaling with object count: increasing the number of objects by a factor of 5 (from 1000 to 5000) increased the runtime by a factor of approximately 6.4 (from 69 to 440 s) while memory usage increased by a factor of 2.3 (from 150 to 350 MB). The sub-linear memory scaling is achieved because the DBSCAN algorithm with spatial indexing structures such as ball-tree or kd-tree has average-case time complexity of $O(n \log n)$ rather than the worst-case $O(n^2)$, and the orbital propagation calculations are embarrassingly parallel across objects.

These performance characteristics indicate that the proposed framework is suitable for operational space situational awareness applications. For typical LEO surveillance scenarios involving hundreds to thousands of detected objects following a breakup event, this method can provide preliminary breakup parameter estimates within minutes on standard workstation hardware, enabling rapid response for event confirmation, debris cloud tracking initiation, and conjunction assessment.

3.6. Analysis of the Intelsat 33E Breakup Event

On October 19, 2024, Intelsat reported the failure of its geosynchronous communication satellite Intelsat 33E. Concurrently, the U.S. Space Force documented the satellite breakup event at 19 October 2024 04:30 UTC via Space-Track.org. SHAO detected four debris (S_1–S_4) during observations conducted on the event day and the subsequent day, with their initial orbital parameters and pre-breakup orbital elements of Intelsat 33E detailed in Table 6.

The breakup analysis using debris (S_1–S_4) required no special temporal parameter adjustments, as their initial orbital epochs were proximate to the breakup time. The clustering parameters were set as: neighborhood radius $\varepsilon_1 = 2000$ km, $\varepsilon_2 = 500$ km, and $MinPts = 4$ (constrained by the limited sample size). As shown in Row 1 of Table 7, these debris formed a cluster through orbit backtracking, reaching maximum density at 19 October 2024 04:43:15 UTC. Within two weeks post-breakup, SHAO identified five additional uncorrelated targets (S_5–S_9) in the original orbital slot, subsequently analyzed as Row 2 in Table 7. Debris

S_1–S_8 formed Cluster 2 with maximum density at 19 October 2024 04:48:02 UTC, while S_9 remained isolated.

Table 6. Orbit elements of Intelsat 33E debris.

Names	Epoch (UTC)	Semi-Major Axis (km)	Inclination (deg)	Eccentricity
Intelsat 33E	2024/10/18 18:10:00	42,165.742	0.085	0.000
S_1	2024/10/19 18:44:00	42,136.629	0.117	0.002
S_2	2024/10/19 17:34:00	42,016.462	0.109	0.003
S_3	2024/10/20 13:21:00	42,288.626	0.391	0.006
S_4	2024/10/20 14:09:00	42,051.726	0.924	0.006
S_5	2024/10/30 14:18:00	41,951.468	0.105	0.005
S_6	2024/10/30 14:37:00	41,678.181	0.350	0.012
S_7	2024/10/30 14:15:00	41,859.601	0.181	0.009
S_8	2024/10/30 14:41:00	41,795.247	0.163	0.009
S_9	2024/10/31 12:33:00	41,816.414	2.656	0.030
S_{10}	2024/11/04 17:19:00	42,021.437	1.878	0.091
S_{11}	2024/11/07 12:48:00	42,045.753	2.441	0.012
S_{12}	2024/11/13 20:03:00	42,117.964	0.133	0.002
S_{13}	2024/11/14 14:18:00	43,209.885	0.128	0.025
S_{14}	2024/11/16 19:52:00	41,111.616	4.758	0.036
S_{15}	2024/11/17 20:41:00	42,145.038	8.147	0.005
S_{16}	2024/11/23 17:02:00	40,518.362	0.853	0.054
S_{17}	2024/12/07 17:45:00	42,721.206	0.344	0.018
S_{18}	2024/12/08 18:49:00	42,063.742	0.760	0.013
S_{19}	2025/01/14 22:18:00	42,749.307	0.400	0.015
S_{20}	2025/01/27 18:10:00	42,391.352	0.409	0.005
S_{21}	2025/02/21 18:08:00	41,698.390	1.918	0.016
S_{22}	2025/02/27 15:44:00	42,770.867	0.536	0.015
S_{23}	2025/02/02 17:28:00	42,402.861	0.407	0.005

Table 7. Satellite Intelsat 33E breakup event analysis result with $\varepsilon_1 = 2000$ km.

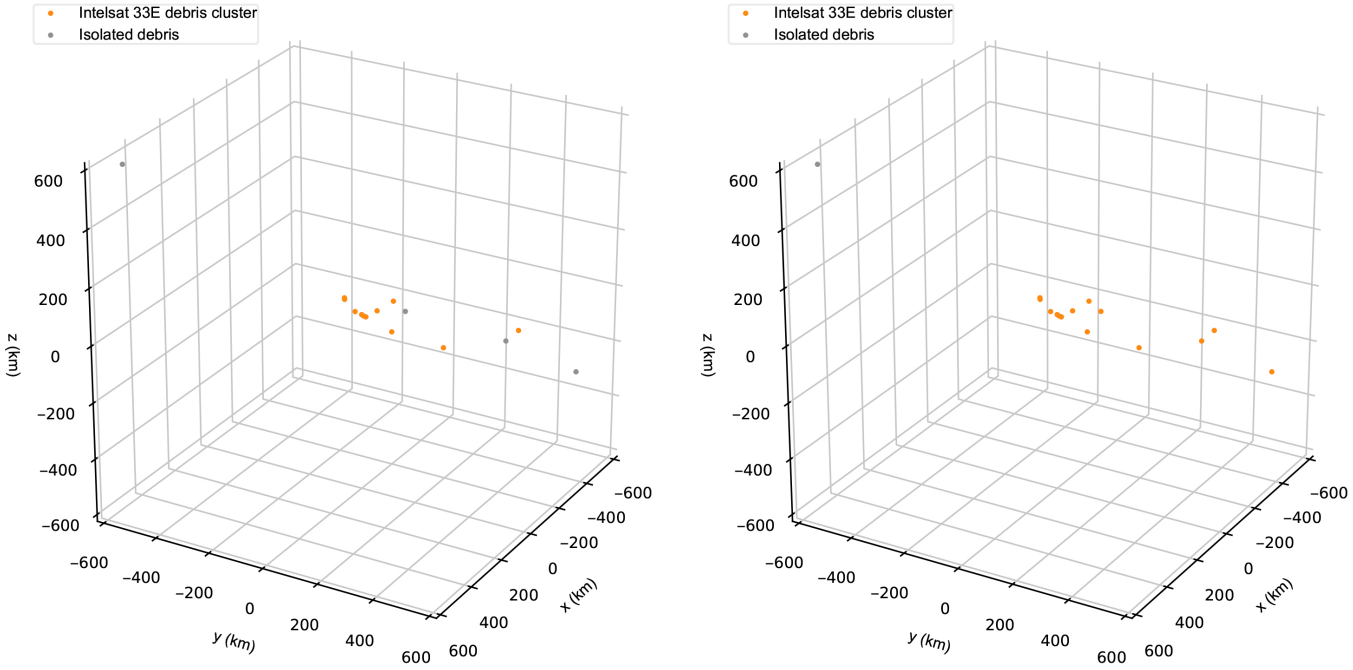
Clusters/Object	Time UTC	Center Position (LLA)	Center Position (GCRS)	Samples
$C_1(S_1$–$S_4)$	2024/10/19 04:43:15	(59.880° E, 0.057° S, 35,774.723 km)	$(-39, 227.891, 15, 428.335, 51.897)$	S_1–S_4
$C_2(S_1$–$S_8)$	2024/10/19 04:48:02	(59.945° E, 0.050° S, 35,784.470 km)	$(-39, 568.224, 14, 561.515, 57.983)$	S_1–S_9
$C_3(S_1$–$S_8, S_{12}, S_{18}, S_{20})$	2024/10/19 04:45:36	(59.814° E, 0.024° S, 35,804.252 km)	$(-39, 394.445, 15, 080.643, 76.174)$	S_1–S_{23}
Intelsat 33E	2024/10/19 04:45:36	(59.953° E, 0.063° S, 35,779.821 km)	$(-39, 408.241, 14, 976.036, 47.902)$	-

By 2025 February 14, new uncorrelated targets (S_{10}–S_{23}) were detected near Intelsat 33E's original position (orbital parameters in Table 6). Full analysis of (S_{10}–S_{23}) (Table 7 Row 3) revealed reformed clusters, with Cluster 3 converged at 19 October 2024 04:45:36 UTC and centered at $(59.814°E, 0.024°S)$—adjacent to Intelsat 33E satellite's operational position. The spatial distribution relative to the original orbit at cluster convergence is visualized in Figure 10a.

Figure 10a analysis identifies three method-classified isolated targets that actually reside within the cluster neighborhood during breakup proximity. These objects were excluded from the clusters generated during the first-layer clustering because they remained isolated (exceeding the Neighborhood Radius 1 from other objects). However, when orbital trajectories were backpropagated to near the breakup epoch, they were found positioned within the neighborhood of the clusters. Two principal factors contribute to this phenomenon:

1. At GEO altitudes around 36,500 km, the impulse generated during satellite breakup events causes debris to rapidly disperse from the original location, resulting in significantly larger spatial separation scales between debris compared to low Earth orbit [22]. Therefore, when applying breakup analysis methods across different orbital regimes, it becomes essential to adapt neighborhood radius parameters accordingly.

2. These debris are not intrinsically isolated. The apparent isolation stems from observational constraints, as Table 5 only contains currently detected suspected breakup debris. Numerous undetected debris likely exist between these targets. Enhanced detection capabilities revealing more debris would enable the DBSCAN algorithm to more effectively establish and propagate neighborhood chaining during clustering. This constitutes another research direction for our subsequent work: conducting targeted searches for potential additional debris clusters within already identified debris regions.



(**a**) (**b**)

Figure 10. Comparison between clusters generated by methods and samples. (**a**) Positions related to Intelsat 33E of Cluster: C_3 at cluster gathering time 19 October 2024 04:45:36 UTC. (**b**) Positions related to Intelsat 33E of Cluster: C_4 at cluster gathering time 19 October 2024 04:38:42 UTC.

When applying the method to larger spatial scales such as GEO, we should employ corresponding parameters tailored to different orbital altitudes. We implemented expanded neighborhood radii $\varepsilon_1 = 3000$ km and $\varepsilon_2 = 500$ km to verify enhanced clustering performance for Intelsat 33E debris. Under these parameters, the clustering results and breakup analysis for debris samples $(S_1–S_{23})$ are presented in Table 8 and Figure 10b. The clustering and satellite breakup analysis reveal that with the 3000 km neighborhood radius at the first clustering layer, three additional targets (S_9, S_{13}, and S_{17}) join the Intelsat 33E debris cluster compared to cluster C_3 in Table 7. As shown in Table 8, cluster C_4 exhibits a convergence time of 19 October 2024 04:38:42 UTC, displaying an 8 min discrepancy from the breakup time reported by the Space-track website, with a 205.05 km separation between the cluster's position at convergence and Intelsat 33E's location at that temporal instant.

Table 8. Satellite Intelsat 33E breakup analysis result with $\varepsilon_1 = 3000$ km.

Clusters/Object	Time UTC	Center Position (LLA)	Center Position (GCRS)
$C_4(S_1–S_9, S_{12}, S_{13}, S_{17}, S_{18}, S_{20})$	2024/10/19 04:38:42	(59.684° E, 0.005° S, 35,818.919 km)	(−38,898.247, 16,355.730, 89.190)
Intelsat 33E	2024/10/19 04:38:42	(59.952° E, 0.063° S, 35,778.009 km)	(−38,938.345, 16,159.192, 46.639)

4. Discussion

This study applies the satellite breakup analysis method proposed in Section 2.2 to investigate two historic events: the February 2009 Cosmos 2251 and Iridium 33 collision breakup and the October 2024 Intelsat 33E breakup. For the Cosmos 2251-Iridium 33 collision case, utilizing pre-identified debris orbital data released by U.S. authorities and prior knowledge of the breakup timing, our analysis demonstrates that the DBSCAN algorithm effectively detects debris clusters with 98.92% clustering accuracy and a 93.20% positive sample recall rate. The derived breakup time (cluster densest moment) shows a 5 s deviation from the predicted collision time and a 21.68 km separation from the predicted collision position of the two satellites' original orbits.

In the Intelsat 33E breakup case, where no prior information was available, the method analyzed observed uncorrelated targets. The results indicate a high probability that Cluster 4 targets in Table 7 represent breakup debris, with calculated breakup time at 19 October 2024 04:38:42 UTC and cluster centroid coinciding with Intelsat 33E's original operational position. The 8 min discrepancy between our derived time and the U.S. Space-Track.org announcement (19 October 2024 04:30 UTC) remains insignificant relative to GEO orbital periods, validating method effectiveness. Notably, Space-Track.org remains the sole public source officially cataloging this breakup time without independent verification.

Over a month elapsed following the breakup event before the United States released the debris catalog via Space-Track.org. The correspondence between the suspected breakup debris identified in our study and the official Space-Track catalog NORAD IDs is presented below: (S1: 41848, the main body of the breakup satellite), (S2: 61992), (S3: 61988), (S5: 61995), (S6: 61991), (S7: 61990), (S8: 61997), (S12: 61998).

4.1. Uncertainty Analysis and Error Sources

The observed discrepancies (5 s temporal, 21.68 km spatial for Iridium-Cosmos; 8 min for Intelsat 33E) originate from multiple sources:

1. TLE accuracy limitations : Public TLEs have typical position uncertainties of 1–10 km, which propagate during backtracking;
2. Fragment Delta-V distribution: Anisotropic fragment velocities affect cluster convergence characteristics.

The 21.68 km spatial error represents approximately 0.3% of the orbital radius, which is acceptable for initial breakup localization in operational response scenarios.

4.2. Discussion Comparing DBSCAN and Other Density-Based Clustering Algorithms

While DBSCAN is employed as the core clustering algorithm in this work, it is instructive to discuss the rationale for choosing DBSCAN over other advanced density-based clustering methods, particularly HDBSCAN [23] and OPTICS [24], which are often considered when cluster densities are non-uniform.

HDBSCAN (hierarchical density-based spatial clustering of applications with noise) extends DBSCAN by automatically extracting clusters from a hierarchy of density-based partitions, thereby eliminating the need to specify the neighborhood radius parameter ε. This is particularly advantageous when dealing with datasets containing clusters of varying densities. However, HDBSCAN requires construction of a minimum spanning tree and hierarchical density tree, followed by cluster extraction from the dendrogram. This multi-step procedure introduces computational overhead compared to DBSCAN's direct ε-neighborhood queries with spatial indexing structures such as ball-tree or kd-tree. For datasets in the range of 10,000 to 50,000 objects—typical for LEO and GEO surveillance applications—DBSCAN provides a faster runtime while maintaining an adequate clustering performance. More importantly, in our operational breakup detection scenario, the primary

objective is to identify whether dense debris clusters exist and to estimate breakup time and location, rather than to further partition a potentially homogeneous debris cloud into density-based sub-clusters. The binary distinction between cluster and background noise suffices for detection purposes, and DBSCAN meets this requirement efficiently.

OPTICS (Ordering Points To Identify the Clustering Structure) produces a reachability ordering that reveals cluster structure at multiple scales, which is conceptually similar to performing DBSCAN with varying ε values. While this provides valuable multi-scale analysis, OPTICS requires a post-processing step to extract actual cluster assignments from the reachability plot, typically using the ξ-clustering method or similar algorithms. This additional extraction step adds complexity for near-real-time operational deployment. Furthermore, the parameters involved in cluster extraction from reachability plots (such as the ξ threshold) do not have the same direct physical interpretation as DBSCAN's ε (spatial distance in kilometers) and MinPts (expected minimum debris count). The interpretation of reachability plots themselves also requires expertise that may not be readily available in operational space situational awareness contexts.

The choice of DBSCAN for this framework is motivated by the specific requirements of operational breakup detection: the need for computational efficiency in near-real-time applications, the sufficiency of binary cluster-versus-noise distinction for detection purposes, and the interpretability of parameters (ε and MinPts) in terms of physical quantities (spatial distance and expected debris count). Future work may explore hybrid approaches or more sophisticated clustering algorithms as the framework evolves to handle increasingly complex scenarios, such as multiple simultaneous breakup events with varying fragmentation intensities.

5. Conclusions

In summary, by employing the method for space debris clustering and satellite breakup event analysis, we successfully reconstructed the scenarios of two distinct satellite breakup events occurring in LEO and GEO, respectively, demonstrating the method's potential applicability across different orbital regimes, though further validation across a broader range of breakup scenarios is needed. The method's principal advantage lies in its ability to systematically analyze uncorrelated targets or debris detected by space surveillance networks without requiring prior knowledge, supporting the identification of potential debris clusters and subsequent characterization of breakup events through cluster analysis. This capability facilitates timely alerts when aggregated debris from breakup events is identified. However, two critical limitations persist in the DBSCAN-based approach: First, the method exhibits heightened sensitivity to the neighborhood radius parameter (ε), necessitating orbit-type-specific threshold selection. Our ongoing research aims to address this through developing multilayer artificial neural networks trained on breakup debris across various orbital regimes to determine optimal parameters. Second, the analysis accuracy degrades when processing debris cataloged long after breakup events or containing low-precision orbital data. Such temporally distant or imprecise orbital elements introduce significant errors in breakup scenario reconstruction, as late-observed debris accumulate propagation uncertainties during orbit backtracking.

Author Contributions: Conceptualization, J.-H.Z. and J.S.; Methodology, J.-H.Z.; Investigation, J.-H.Z. and J.S.; Formal analysis, J.-H.Z. and J.S.; Software, Q.L.; Validation, J.-H.Z., J.S. and Q.L.; Resources, Y.-D.M.; Writing—original draft preparation: J.-H.Z.; Writing—review and editing, J.-H.Z. and Y.-D.M.; Visualization, J.-H.Z.; Supervision, Y.-D.M. and J.S.; Project administration, Y.-D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China of funder grant number 2025YFE0107000.

Data Availability Statement: The original data presented in the study are openly available at https://pan.baidu.com/s/1Y2fpWK94LeZg7vK5clPjhQ?pwd=jfak, accessed on 21 February 2026.

Acknowledgments: Thanks to all the researchers and engineers of the Shanghai Astronomical Observatory who participated in this job.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DBSCAN	Density-Based Spatial Clustering of Applications with noise
RSO	Resident Space Objects
ESA	European Space Agency
LEO	Low Earth Orbit
GEO	Geosynchronous orbit
SHAO	Shanghai Astronomical Observatory
SSA	Space Situational Awareness
LLA	Longitude, Latitude, Altitude
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
OPTICS	Ordering Points To Identify the Clustering Structure

References

1. Agency, E.S. Space Debris by the Numbers. 2025. Available online: https://www.esa.int/Space_Safety/Space_Debris/Space_debris_by_the_numbers (accessed on 12 April 2026).
2. Office, E.S.D. *ESA's Annual Space Environment Report*; Technical report; European Space Agency: Darmstadt, Germany, 2025.
3. Pardini, C.; Anselmo, L. Revisiting the collision risk with cataloged objects for the Iridium and COSMO-SkyMed satellite constellations. *Acta Astronaut.* **2017**, *134*, 23–32. [CrossRef]
4. Kessler, D.J.; Cour-Palais, B.G. Collision frequency of artificial satellites: The creation of a debris belt. *J. Geophys. Res.* **1978**, *83*, 2637–2646. [CrossRef]
5. Rex, D.; Eichler, P.; Soppa, U.; Zuschlag, J.; Bade, A. Space debris—Origin, evolution and collision mechanics. *Acta Astronaut.* **1989**, *20*, 209–216. [CrossRef]
6. Rossi, A.; Cordelli, A.; Farinella, P.; Anselmo, L.; Pardini, C. Long term evolution of the space debris population. *Adv. Space Res.* **1997**, *19*, 331–340. [CrossRef]
7. Ravi, P.; Black, A.; Frueh, C. Investigation of Atlas V Centaur Upper Stage Fragmentation Events. *J. Astronaut. Sci.* **2023**, *70*, 12. [CrossRef]
8. Dimare, L.; Cicalo, S.; Rossi, A.; Alessi, E.M.; Valsecchi, G.B. In-orbit fragmentation characterization and parent bodies identification by means of orbital distances. In Proceedings of the First International Orbital Debris Conference, Sugarland, TX, USA, 9–12 December 2019.
9. Mains, D.; Peterson, G.; McVey, J.; Maldonado, J.; Sorge, M. Forensic analysis of recent debris-generating events. In Proceedings of the Second International Orbital Debris Conference, Sugarland, TX, USA, 4–7 December 2023.
10. Montaruli, M.F.; Lizia, P.D.; Cordelli, E.; Ma, H.; Siminski, J. A stochastic approach to detect fragmentation epoch from a single fragment orbit determination. *Adv. Space Res.* **2023**, *72*, 3713–3733. [CrossRef]
11. Mains, D.; Sorge, M.; Cabrera, V. Techniques for quick-response representations of fragmentation events. In Proceedings of the Second International Orbital Debris Conference, Sugarland, TX, USA, 4–7 December 2023.
12. Muciaccia, A.; Romano, M.; Trisolini, M.; Colombo, C. Reconstruction of in-orbit breakup events over the long term. *Acta Astronaut.* **2025**, *228*, 875–885. [CrossRef]
13. Grattagliano, P.; Mignocchi, A.; Montaruli, M.F.; Lizia, P.D.; Cecco, A.D.; Castronuovo, M.M. Early stage characterization of on-orbit fragmentation events. *Acta Astronaut.* **2025**, *236*, 32–46. [CrossRef]
14. Wu, D.; Rosengren, A.J. An investigation on space debris of unknown origin using proper elements and neural networks. *Celest. Mech. Dyn. Astron.* **2023**, *135*, 44. [CrossRef]

15. Jiang, Y. The Situation and Evolution of Two Debris Clouds Generated by the Collision of Two Satellites. *Space Debris Res.* **2023**, *23*.
16. Giudici, L.; Colombo, C.; Horstmann, A.; Letizia, F.; Lemmens, S. Density-based evolutionary model of the space debris environment in low-Earth orbit. *Acta Astronaut.* **2024**, *219*, 115–127. [CrossRef]
17. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*; AAAI Press: Washington, DC, USA, 1996.
18. Vallado, D.; Crawford, P.; Hujsak, R. Revisiting Spacetrack Report #3. In Proceedings of the AIAAAAS Astrodynamics Specialist Conference, Keystone, CO, USA, 21–24 August 2006.
19. Luzum, B.J.; Petit, G. The IERS Conventions (2010): Reference systems and new models. In Proceedings of the International Astronomical Union, Beijing, China, 20–31 August 2012; pp. 227–228.
20. Bowring, B.R. Transformation from spatial to geographical coordinates. *Surv. Rev.* **1976**, *23*, 323–327. [CrossRef]
21. Kelso, T. Analysis of the Iridium 33-Cosmos 2251 Collision. In Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, USA, 1–4 September 2009.
22. Friesen, L.J.; Jackson, A.A.; Zook, H.A.; Kessler, D.J. Results in orbital evolution of objects in the geosynchronous region. *J. Guid. Control Dyn.* **1992**, *15*, 263–267. [CrossRef] [PubMed]
23. McInnes, L.; Healy, J.; Astels, S. Hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [CrossRef]
24. Ankerst, M.; Breunig, M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.