A Transformer-Based Framework for Generalized Maneuver Detection and Reconstruction

Matteo Rosa
Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, Milano, Italy, matteo.rosa@polimi.it
Gaetano Calabrò
Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, Milano, Italy, gaetano.calabro@polimi.it
Pierluigi Di Lizia
Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, Milano, Italy, pierluigi.dilizia@polimi.it

ABSTRACT

The rapid growth of the Low Earth Orbit (LEO) environment has increased the operational complexity of Space Situational Awareness (SSA) and Space Domain Awareness (SDA), for which reliable detection of satellite maneuvers from publicly available Two-Line Element (TLE) data remains a challenging problem. Recurrent architectures, and Long Short-Term Memory (LSTM) networks in particular, detect maneuvers accurately when trained and tested on the same spacecraft, but their performance degrades on previously unseen satellites, limiting their operational scalability. This paper presents a supervised Transformer-based framework addressing both generalization and maneuver reconstruction. The model couples a multi-head self-attention encoder to a convolutional decoder and processes the semi-major axis and the inclination in two structurally identical but independent branches, so that in-plane and out-of-plane maneuvers are detected separately. A gradient maneuver mask replaces the conventional binary labeling: time steps within a maneuver interval receive a high value that decays gradually outside it, so that the learning objective becomes the reconstruction of the temporal extent of the maneuver rather than the detection of an isolated event. The model is evaluated in two scenarios, on satellites included in the training set and on previously unseen spacecraft, reaching F_1-scores of approximately 0.95 in both cases without retraining the model on each satellite, and correctly reconstructing more than 90% of the detected maneuvers at an Intersection over Union (IoU) above 0.5. The trained network is finally applied to satellites with no publicly available maneuver history, reconstructing their maneuver catalogs from TLE data alone.

1. INTRODUCTION

The increasing congestion of the near-Earth environment has made orbital safety and sustainability a central concern for the space community. In this context, Space Situational Awareness (SSA) and Space Surveillance and Tracking (SST) play a fundamental role in enabling safe operations in orbit [1]. A key aspect of these domains is the continuous monitoring and prediction of space object trajectories, with particular focus on the detection of orbital maneuvers. Accurate maneuver detection is essential for maintaining reliable catalogs, improving conjunction assessment accuracy, and supporting timely collision avoidance decisions.

The problem of maneuver detection has become a central topic in the SSA domain in recent years, and many approaches have been proposed in the literature. Existing methods can generally be classified into two main categories: measurement-based and data-driven approaches. Measurement-based methods rely on tracking measurements, which are not always publicly available and may be accessible only to specific operators. On the other hand, data-driven methods exploit publicly available data sources, such as Two-Line Element (TLE) sets, making them applicable in a broader range of scenarios. This work contributes to the advancement of the state of the art in the field of data-driven maneuver detection.

Data-driven methods reported in the literature range from statistical approaches to techniques based on machine learning and deep learning. Among the earliest studies in the field of statistical methods, Kelecy et al. [2] proposed an

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

approach aimed at identifying abrupt changes in orbital elements. Their method detects maneuvers from TLE time series by first smoothing orbital elements, such as the semi-major axis, orbital energy, and inclination, using polynomial filters and then comparing adjacent data segments. A maneuver is detected when the differences between consecutive smoothed segments exceed a predefined threshold, indicating a significant change in the orbital parameters. The main limitation of this method lies in the selection of the detection threshold, which must be carefully tuned for each satellite to achieve reliable performance. Subsequently, Patera [3] proposed another statistical approach for maneuver detection based on the analysis of orbital parameter time series. This method uses moving windows over sequential batches of data, within which a polynomial is fitted to filter out noise. Maneuvers are flagged if the deviation from the fit exceeds a predefined threshold. As with previous statistical approaches, detection performance is highly sensitive to noise and requires careful threshold tuning. These limitations were partially addressed by Lemmens et al. [4], who proposed an adaptive threshold approach in which the detection threshold is automatically updated based on the statistical properties of the data itself. The research focus subsequently shifted from statistical approaches toward machine learning techniques. Early studies explored conventional machine learning methods based on clustering algorithms for maneuver identification, while more recent works have investigated deep learning approaches, particularly those that leverage Long Short-Term Memory (LSTM) networks to capture temporal dependencies in orbital time-series data. Following this research direction, Bai et al. [5] formulated maneuver detection as an unsupervised data mining problem by applying clustering algorithms, such as K-means, hierarchical clustering, and fuzzy C-means, to orbital features derived from TLE data. In their approach, maneuvers are identified as transitions between clusters representing different orbital behavior regimes, avoiding the need for predefined detection thresholds. Similarly, Mital et al. [6] demonstrated how both supervised and unsupervised learning techniques can be employed to identify the characteristics of a target that are most relevant for change detection, effectively encoding its geospatial behavior over time. However, clustering-based approaches are inherently limited in their ability to capture temporal dependencies among orbital parameters. To address this limitation, the authors proposed the use of deep learning architectures, such as Recurrent Neural Networks (RNNs) and LSTM networks, which are specifically designed to model sequential data. Their work represented a starting point for the exploration of deep learning techniques for satellite behavior modeling and maneuver detection, opening a new research direction in the field. DiBona et al. [7] proposed a maneuver classification framework based on a neural network capable of producing Boltzmann-distributed posterior probabilities, thereby providing a measure of confidence across the different maneuver classes. The considered maneuver classes correspond to those most commonly encountered during the operational lifetime of a satellite, including North–South station-keeping, East–West station-keeping, transfer maneuvers, and end-of-life maneuvers. Roberts et al. [8] formulated Geostationary orbit (GEO) maneuver detection as a supervised learning problem using Convolutional Neural Networks (CNNs) trained on labeled time series derived from TLE data and transformed into geographic coordinates. The proposed model learns characteristic patterns associated with longitudinal shift maneuvers, enabling not only maneuver classification but also orbital anomaly detection through the identification of deviations from the pattern of life of the satellite. Cipollone et al. [9] [10] proposed a maneuver detection approach based on LSTM networks to learn the operational pattern of life of a satellite from its historical TLE data. Leveraging prior knowledge of the maneuver history of a satellite, each TLE was labeled as either maneuvering or non-maneuvering, enabling the supervised training of the model. The results demonstrated excellent performance when the model was evaluated on the same satellite used during training. However, a major limitation of this approach concerns its ability to generalize across different satellites. In particular, the detection performance degrades significantly when the trained model is applied to satellites that were not included in the training dataset. This represents a significant challenge for supervised learning approaches, as publicly available data describing satellite maneuvers are limited. To address this issue, Cipollone et al. [11] proposed an unsupervised approach based on a Bidirectional LSTM autoencoder. In this setting, the model is trained exclusively on ballistic debris trajectories, where no maneuvers are assumed to occur, allowing it to learn nominal ballistic behavior. During inference, significant deviations from the reconstructed patterns are flagged as potential maneuvers. This approach shows promising results in terms of generalization; however, its main limitation lies in the careful selection of debris used for training, which must exhibit orbital characteristics sufficiently similar to those of the satellites on which the model is applied.

The reviewed studies show that the problem of maneuver detection has been extensively explored in the literature. The main challenges emerging from these studies are, on the one hand, the scarcity of maneuver histories available for constructing labeled datasets for supervised detection techniques, and, on the other hand, the difficulty of developing models capable of generalizing maneuver detection to satellites not included in the training set. This latter issue is particularly significant in supervised approaches, while in unsupervised methods it has also been observed that

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

generalization performance strongly depends on the appropriate selection of debris orbits used to train the model.

This work aims to address the generalization limitations of existing maneuver detection methods by developing a model capable of reconstructing the maneuver history of satellites for which such information is not publicly available. This approach could also help overcome the limited availability of publicly accessible maneuver data. The maneuver histories reconstructed by the proposed model can be used as additional training data, enabling the model to be retrained on a larger dataset and potentially improving its generalization capability. To achieve this objective, the proposed approach exploits the ability of Transformer architectures to model temporal sequences and learn complex behavioral patterns through a network composed of a Transformer-based encoder and a convolutional decoder. The model is trained and evaluated only on Low Earth Orbit (LEO) satellites, because the 16 satellites for which the International Laser Ranging Service (ILRS) database provides maneuver histories are all in LEO. This paper is organized as follows. After the introduction, Section 2 introduces the theoretical background of Transformer architectures, with particular emphasis on the attention mechanism. Section 3 presents the proposed method, describing the complete pipeline from data pre-processing to the design of the neural network. Section 4 reports the experimental results obtained using the testing pipeline and illustrates the output produced by the operational pipeline. In addition, it shows the comparison of the proposed approach with a classical statistical detection method. Finally, Section 5 concludes the paper discussing the results and outlining directions for future work.

2. FUNDAMENTALS

Transformers represent the current state of the art in deep learning for modeling sequential data. Originally introduced in the field of Natural Language Processing (NLP) [12], they have subsequently been successfully applied to time-series analysis [13]. Transformers are based on the attention mechanism, which captures dependencies among input features without relying on recurrent architectures, as is the case with RNNs. Through the attention mechanism, input data are processed in parallel, allowing the model to capture both local and global dependencies among the parameters.

The attention function can be described as a mapping from a query and a set of key-value pairs to an output. The attention mechanism introduced in [12], known as scaled dot-product attention, computes the dot product between the query and each key, divides the resulting scores by $\sqrt{d_k}$, and applies a softmax function to obtain the attention weights. These weights are then used to compute a weighted combination of the corresponding values:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

Rather than relying on a single attention function, Transformers employ multiple attention heads by linearly projecting the queries, keys, and values into h different representation subspaces. These projections are computed in parallel, and the resulting d_v-dimensional outputs are concatenated and linearly projected once again to produce the final representation. This mechanism, known as multi-head attention, enables the model to attend to information from multiple representation subspaces simultaneously, allowing it to capture different types of relationships within the input sequence.

$$\text{MultiHead}(Q,K,V) = \text{Concat}\left(\text{head}_1,\ldots,\text{head}_h\right)W^O, \quad \textit{where}\ \text{head}_j = \text{Attention}\left(QW_j^Q, KW_j^K, VW_j^V\right) \tag{2}$$

In addition to the attention layers, the Transformer architecture includes a fully connected feed-forward network (FFN), consisting of two linear transformations with a ReLU activation function in between.

$$\text{FFN(x)} = max(0, xW_1 + b_1)W_2 + b_2 \tag{3}$$

Transformers do not encode the order of the input sequence. In order to make the model aware of the position of each token, positional information is injected into the input embeddings through a positional encoding. In the original Transformer architecture, positional encoding is implemented using sine and cosine functions whose wavelengths form a geometric progression spanning a factor of 10^4, so that the encoding depends on the position of the token within the sequence:

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i})} = sin(pos/10000^{2i/d_{model}}) \qquad \mathrm{PE}_{(\mathrm{pos},2\mathrm{i}+1)} = cos(pos/10000^{2i/d_{model}}) \tag{4}$$

The formulation above describes the standard Transformer architecture originally proposed for NLP [12]. In the proposed framework, four components are adapted to the characteristics of TLE time series: the input embedding, the activation function of the FFN, the temporal encoding, and the decoder, as described in Section 3.4.

3. METHODOLOGY

The overall workflow of the proposed approach is schematically illustrated in Fig. 1. The framework consists of three main blocks: Data Pre-processing, Transformer-Based Neural Network, and Maneuver Detection and Performance Evaluation. The pipeline begins with a data pre-processing stage, where the raw TLE data are converted into input time series suitable for the neural network. The processed sequences are then fed to the Transformer-based model, which predicts whether each sample belongs to a maneuver or a non-maneuver interval. Finally, the network outputs are post-processed to identify complete maneuver intervals and, when ground-truth labels are available, to evaluate the model performance.

As shown in Fig. 1, the proposed framework includes two different pipelines. The *Training and Testing Pipeline* describes the workflow used to train and evaluate the model. In this case, only satellites with known maneuver histories are considered, enabling the generation of ground-truth labels for supervised training and performance evaluation. The *Operational Pipeline*, on the other hand, illustrates the deployment of the trained model to reconstruct the maneuver history of satellites for which no public maneuver records are available, using only their historical TLE data.

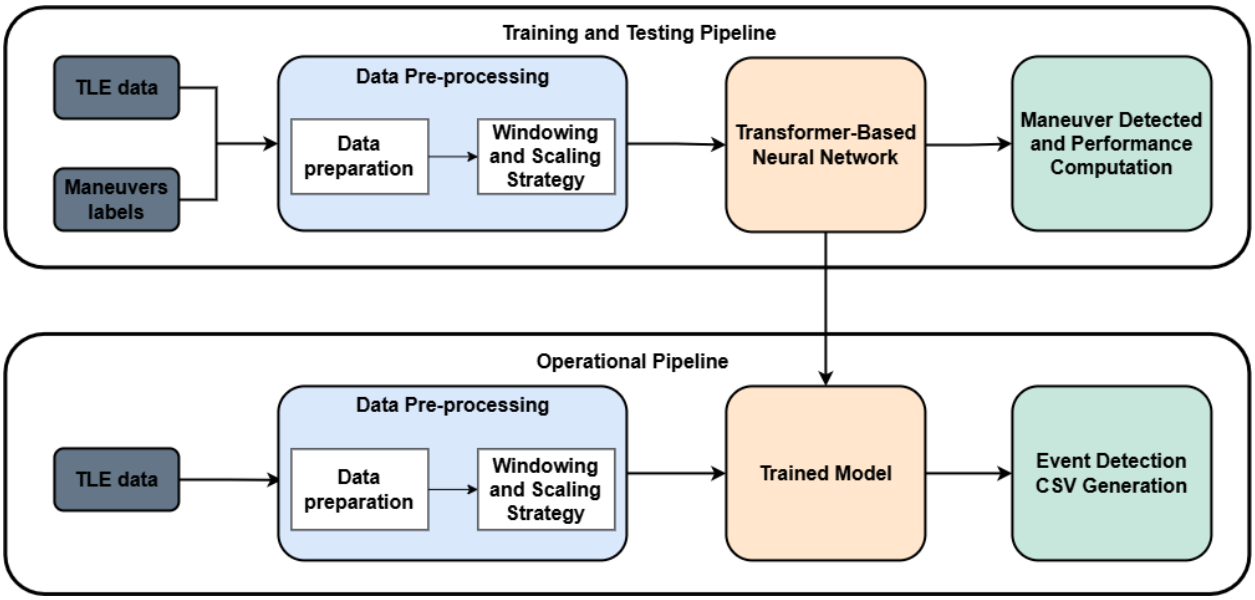


Fig. 1: Overview of the Maneuver Detection pipeline. It consists of a training and testing pipeline, together with an operational pipeline.

3.1 TLE Data Preparation

The proposed pipeline relies exclusively on publicly available data, specifically historical TLEs, which are regularly published and updated on Space-Track. However, Space-Track does not provide information regarding satellite maneuver histories. To address this limitation, maneuver data were obtained from the ILRS, which provides, for a limited number of satellites, maneuver start and end dates as well as the corresponding ΔV values. For the training, validation, and testing phases, TLE data were collected exclusively for satellites whose maneuver histories are available through ILRS. This enabled the construction of a reliable ground truth, which was used both to train the model in a supervised way and to evaluate its performance. To assess the operational applicability of the proposed approach, TLE data from satellites with no publicly available maneuver history were also analyzed, thereby reproducing a realistic operational

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

scenario in which maneuver information is unavailable. The complete list of satellites with known maneuver history is reported in Table 1. The table also reports the number of semi-major axis maneuvers (#Man sma) and inclination maneuvers (#Man inc).

Table 1: Satellite Dataset

Satellite	Start Date	End Date	#Man sma	#Man inc	Altitude (km)	Inclination (deg)
CryoSat-2	2010-08-17	2020-05-17	112	0	~ 717	~ 92
Envisat	2002-04-10	2010-09-04	126	31	~ 790	~ 98.55
Haiyang-2A	2011-10-06	2016-02-26	31	0	~ 983	~ 99.3
Haiyang-2C	2020-10-19	2024-08-05	30	0	~ 952	~ 99.3
Haiyang-2D	2021-06-22	2024-08-25	27	0	~ 952	~ 99.3
Jason-1	2002-05-27	2012-03-28	62	0	~ 1336	~ 66
Jason-2	2008-08-05	2016-08-20	34	0	~ 1336	~ 66
Jason-3	2016-02-27	2024-08-25	44	0	~ 1336	~ 66
Saral	2013-06-13	2023-09-30	52	3	~ 800	~ 98.55
Sentinel-3A	2016-04-18	2024-08-25	96	26	~ 815	~ 98.65
Sentinel-3B	2018-12-05	2024-08-25	73	18	~ 814	~ 98.65
Sentinel-6A	2020-12-29	2026-07-15	21	0	~ 1336	~ 66
SPOT-2	1992-08-02	2009-04-20	107	10	~ 832	~ 98.7
SPOT-4	1998-07-13	2012-08-30	104	10	~ 832	~ 98.7
SPOT-5	2002-05-21	2015-02-20	96	7	~ 832	~ 98.7
SWOT	2023-07-26	2026-07-07	38	0	~ 857	~ 77.6

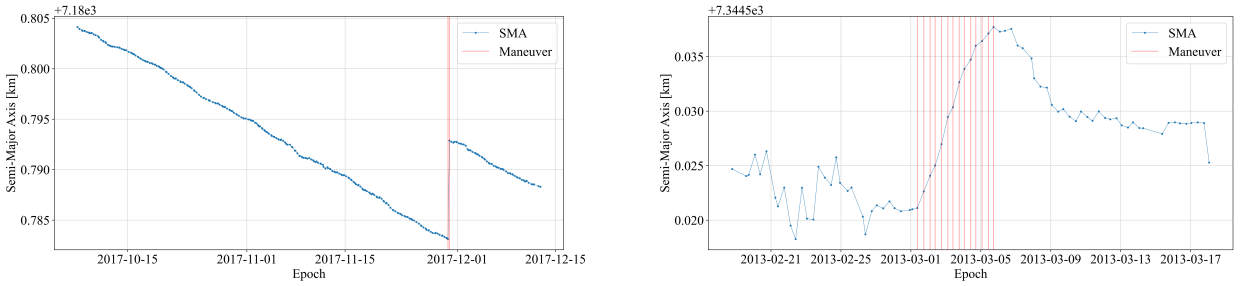
The full set of six Keplerian orbital elements is extracted from the TLEs. In this work, however, only the semi-major axis and inclination are considered, as they are the primary parameters characterizing in-plane and out-of-plane maneuvers for LEO satellites. The resulting time series of orbital elements are clipped to remove the Launch and Early Orbit Phase (LEOP) as well as the decommissioning phase. The orbital parameter time series are used without applying either resampling or smoothing, unlike several existing approaches [10]. This choice is motivated by the objective of allowing the model to learn the noise characteristics directly from the data and to distinguish them from actual maneuver signatures. Furthermore, Transformers do not require uniformly sampled input sequences. When the data are irregularly sampled, the time between consecutive observations can be provided to the model, for example through a time encoding based on the actual observation times.

3.2 Maneuver Label Preparation

As described previously, maneuver information is retrieved from ILRS. The maneuver epochs are associated with the corresponding TLEs according to their timestamps. Rather than assigning the maneuver label to a single TLE, the proposed approach aims to construct a ground truth that captures the entire variation induced by the maneuver on the orbital parameter. Starting from the TLE associated with the maneuver epoch, a maneuver interval is therefore defined by identifying a pre-maneuver TLE and a post-maneuver TLE, as illustrated in Fig. 2. The maneuver intervals shown in Fig. 2 illustrate the two cases encountered in the dataset. Case (a) corresponds to a maneuver involving only two TLEs, resulting in an abrupt jump in the orbital parameter time series. In contrast, case (b) shows a maneuver spanning multiple TLEs, where the variation is distributed over several consecutive epochs and the overall maneuver is represented by the cumulative effect of a series of smaller changes. Because maneuver signatures may be affected by noise and occasional spikes, the pre- and post-maneuver labels initially obtained from the labeling procedure are manually refined. Each maneuver interval is visually inspected, and whenever the initial boundaries do not accurately represent the maneuver, the labels are manually corrected. This refinement ensures that the ground-truth maneuver intervals are as accurate as possible.

For each orbital parameter, a maneuver mask is then generated and used as the ground truth during both the training and testing phases. Unlike the conventional one-hot mask adopted in [10], where only the TLE associated with the maneuver is assigned a value of 1 while all the remaining samples are assigned 0, the proposed method employs a gradient mask. All samples within the maneuver interval, bounded by the pre-maneuver and post-maneuver TLEs, are assigned a value of 1, whereas non-maneuver samples are assigned a value of 0. Moreover, instead of an abrupt transition between maneuver and non-maneuver regions, the mask incorporates a smooth exponential decay at the

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com



(a) Maneuver composed of two TLEs

(b) Maneuver composed of multiple TLEs

Fig. 2: Definition of maneuvers in terms of pre- and post-maneuver TLE.

interval boundaries. Specifically, the transition from 1 to 0 is modeled using an exponential decay with a decay rate of 0.7 until reaching a value of 0.16 [14]. This smoother transition provides the network with a more gradual training signal, making the learning process easier while preserving the primary objective of accurately identifying maneuver regions rather than the transition zones themselves.

3.3 Windowing and Scaling Strategy

Once the maneuver labels and orbital parameter time series have been generated, the data for each satellite are first divided into the training, validation, and test sets. Each set is then partitioned into overlapping windows, whose length and stride are reported in the results section. It is important to perform the dataset split before window generation. Otherwise, overlapping windows could cause the same maneuver to appear simultaneously in different subsets, creating data leakage which would result in an overestimation of the model performance. The use of overlapping windows is intended to generate multiple views of the same maneuver, allowing the model to observe it under different temporal contexts. This strategy effectively increases the number of training samples without introducing additional maneuver events, partially mitigating the limited availability of labeled maneuvers for supervised learning.

Due to the difference in altitude among satellites, as reported in Table 1, it is necessary to rescale the parameters to a common value range. First, each satellite time series is centered around zero. This is achieved by subtracting the median value computed within each window into which the time series is divided. The median is computed per window rather than globally in order to avoid cases in which large changes in semi-major axis or long-term decay phases could bias the centering process. The same procedure is applied to the inclination, which is also reported in Table 1. After centering, the time series are concatenated to form the complete training, validation, and test sets. Finally, the data are scaled by fitting a RobustScaler on the training set and applying the same transformation to both validation and test sets. The resulting scaling parameters are stored and reused in the operational pipeline, where maneuver histories must be reconstructed.

3.4 Network Architecture

The network takes as input the pre-processed orbital parameter sequences described in the previous sections. During training, the maneuver masks provide the labels used to train the model and are not required at inference, allowing the trained detector to be applied to satellites whose maneuver history is unknown. Its objective is to produce, for each input window, a score for every time step representing the confidence that the corresponding orbital parameter value belongs to a maneuver performed by the satellite. The model maps a window $X \in R^{T \times F}$ of $T = 512$ consecutive orbital element values over F channels, together with the associated epoch vector $\tau \in R^{T}$, to a score field $\hat{Y} \in [0,1]^{T \times F}$. In this configuration $F = 2$, corresponding to semi-major axis and inclination. The architecture couples a Transformer-based encoder to a convolutional decoder. The encoder maps the input window into a latent representation in which the attention mechanism relates every time step to every other one within the window, while the decoder maps that representation back onto a score for each time step. The overall data flow is summarized in Fig. 3. The F channels are processed by structurally identical but independent streams: embedding, self-attention and decoding are all specific for each element, and the model produces one score field per orbital element.

Each input stream consists of a scalar time series $x \in R^{T \times 1}$, which is projected to the model dimension through a one-dimensional convolutional layer with a stream-specific kernel size k_f. The convolution uses unit stride and same

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

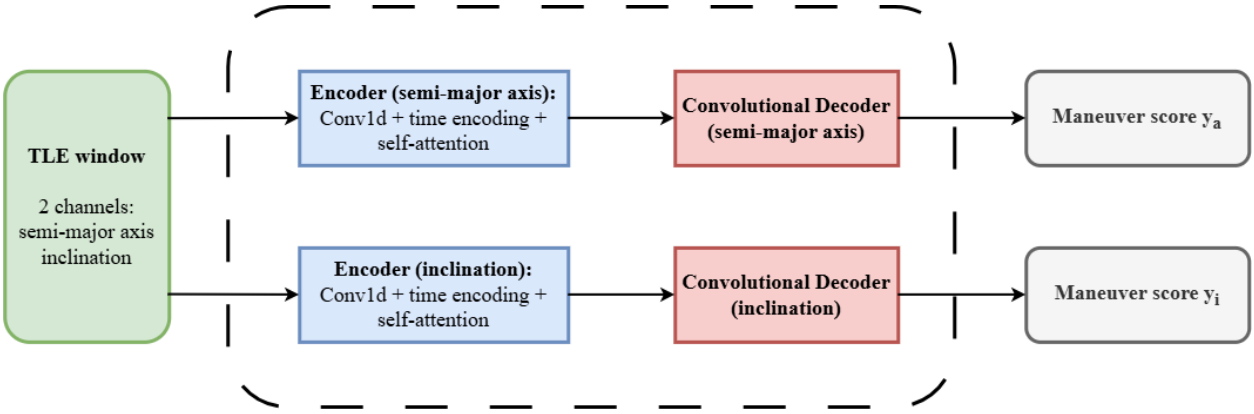


Fig. 3: Neural network architecture.

padding, so that the sequence length is preserved. This choice is dictated by the nature of the signal to be detected: a maneuver does not appear as an anomalous individual value but as a variation distributed over a few consecutive TLEs, and its identification therefore requires the comparison of neighboring samples. A pointwise linear projection would associate each token with a single sample. The representation would then carry no information on the local shape of the signal, and self-attention would compare instantaneous values only [15]. The convolutional embedding instead associates each token with a neighborhood of k_f TLEs, so that the variation induced by the maneuver is encoded within the token itself before the sequence is processed by the Transformer encoder. The same solution is adopted in Transformer architectures for time-series forecasting, in which the input values are projected to the model dimension through a one-dimensional convolution rather than through a linear layer [16]. Since the proposed model operates directly on the irregularly sampled TLE sequence, token indices do not accurately represent the temporal spacing between consecutive observations. For this reason, positional information is encoded using the true elapsed time associated with each TLE rather than its position in the sequence. Specifically, a sinusoidal temporal encoding is computed as

$$\mathrm{TE}[n,2k] = \sin(\omega_k t_n), \qquad \mathrm{TE}[n,2k+1] = \cos(\omega_k t_n) \tag{5}$$

where t_n is the elapsed time from the beginning of the input window and ω_k are learnable frequencies optimized together with the network parameters. The wavelengths are initialized log-spaced over $[\lambda_{\min}, \lambda_{\max}] = [\,2\,\Delta t_{\mathrm{med}},\, 2\,T\,\Delta t_{\mathrm{med}}\,]$, where Δt_{med} is the median gap between TLEs of the training set. This replaces the conventional geometric progression spanning a factor of 10^4, which is calibrated for integer token indices. When the same progression is applied to elapsed time expressed in days, the highest-frequency channels can alias and the lowest-frequency ones become nearly constant. The temporal encoding is added to the convolutional embedding, and layer normalization is applied to the resulting sum before entering the Transformer encoder. The temporal encoding is the only component shared between the two streams. Each stream is processed by a stack of L Transformer encoder blocks. Every block consists of bidirectional multi-head self-attention followed by a position-wise feed-forward network with GELU activation, with residual connections and layer normalization applied around each sublayer. The attention mechanism operates over the entire input window, allowing the model to capture both local and long-range temporal dependencies and to identify maneuvers by analyzing the orbital evolution before and after each event. The decoder consists of N one-dimensional convolutional layers. Each layer includes a 1D convolution, group normalization, a GELU activation function, and dropout. The number of channels is progressively reduced throughout the decoder, and a final 1×1 convolution maps the latent representation to a single output channel. A sigmoid activation is then applied to produce a maneuver score in the range $[0,1]$. A convolutional decoder is adopted instead of a Multi-Layer Perceptron (MLP), since an MLP predicts each time step independently. In contrast, the convolutional layers exploit the local temporal context, allowing each prediction to depend not only on the corresponding encoder output but also on its neighboring representations. This encourages smoother and more consistent predictions over maneuver intervals, improving the reconstruction of the target masks.

3.5 Training Setup

The network is trained with a binary cross-entropy (BCE) loss, computed independently for each orbital parameter and averaged over the two streams. Since maneuver time steps represent only a small fraction of each sequence, the

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

loss is weighted to prevent it from being dominated by the non-maneuver class. This produces class weights that assign greater importance to the minority class (maneuver samples) and lower importance to the majority class (non-maneuver samples), thereby mitigating the class imbalance problem. BCE is preferred over a mean squared error because the mask is close to either 0 or 1 over the vast majority of the time steps, so the sigmoid output operates in its saturated regime. With a mean squared error the gradient with respect to the pre-activation is scaled by the derivative of the sigmoid. This derivative vanishes in the saturated regime, so confidently incorrect predictions produce almost no learning signal. With binary cross-entropy this factor cancels exactly and the gradient reduces to the residual between prediction and target.

Optimization is performed with AdamW [17] and a one-cycle learning-rate schedule [18] with cosine annealing, with gradients clipped to a maximum norm of 1. Training runs for at most 90 epochs and is stopped when the validation selection metric does not improve for 15 consecutive epochs. The selection metric is the harmonic mean of the per-parameter F_1-scores defined in Section 4.1, also used as the hyperparameter optimization objective, and the reported model is the checkpoint of the best epoch rather than the last one.

4. RESULTS

4.1 Experimental Setup

The performance of the proposed maneuver detection method is evaluated using the satellites with known maneuver histories presented in Table 1. Among the 16 satellites considered, 10 are used for training the neural network, while the remaining 6 are reserved exclusively for testing. In the latter case, the model is evaluated on the complete TLE history of each satellite, thereby simulating the operational deployment scenario in which the maneuver history is unknown. The splitting of satellites between the training and only test sets is reported in Table 2.

Table 2: Split of the satellites into training and only test sets.

Training and Testing Satellites	Only Test Satellites
CryoSat-2	Jason-3
Envisat	Sentinel-3B
Haiyang-2C	Sentinel-6A
Haiyang-2D	SPOT-2
Jason-1	SWOT
Jason-2	Haiyang-2A
Saral	
Sentinel-3A	
SPOT-4	
SPOT-5	

For the satellites used during training, the orbital parameter time series are divided into training, validation, and test sets following a 70%/15%/15% split of each complete trajectory history of the satellite: 70% of the samples are assigned to the training set, 15% to the validation set, and the remaining 15% to the test set. The resulting datasets are then partitioned into overlapping windows of 512 samples using a stride of 1.

The neural network hyperparameters were optimized using Optuna [19], a framework for Bayesian hyperparameter optimization. Given an objective function and a predefined search space, Optuna iteratively explores candidate hyperparameter configurations, progressively focusing on the most promising regions of the search space based on the outcomes of previous trials. In this work, the objective function to be minimized was defined based on the model performance on the two orbital parameters considered. Specifically, the objective function is defined as follows:

$$\mathrm{OF} = 1 - \frac{2}{\frac{1}{F_{1,sma}} + \frac{1}{F_{1,inc}}} \tag{6}$$

where $F_{1,\mathrm{sma}}$ and $F_{1,\mathrm{inc}}$ denote the F_1-scores obtained for maneuver detection based on the semi-major axis and inclination, respectively. In this case, the F_1-scores are computed at the level of the individual sample rather than at the level

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

of the maneuver event: every time step of every validation window contributes to the score, after thresholding both the predicted score and the ground truth at 0.9. This metric drives hyperparameter selection, while the detection metrics reported in Section 4.2 are computed at the event level. The objective function is formulated using the harmonic mean of the two scores rather than their arithmetic mean, as the harmonic mean penalizes configurations in which one score is significantly lower than the other. Consequently, the optimization process selects hyperparameter configurations that achieve balanced performance on both orbital parameters, preventing high accuracy on one parameter from compensating for poor performance on the other. Table 3 summarizes the hyperparameters included in the optimization procedure together with the corresponding search space defined for Optuna.

Table 3: Hyperparameters search space.

Hyperparameter	Type	Range/Set
Model dimension	categorical	$[64, 128, 256]$
Head dimension	categorical	$[16, 32, 64]$
Number of layers encoder	integer	$[1, 4]$
Dropout encoder	float	$[0.05, 0.50]$, step 0.05
Encoder kernel size sma	categorical	$[5, 7, 9, 13, 15]$
Encoder kernel size inc	categorical	$[5, 7, 9, 13, 15]$
Decoder kernel size sma	categorical	$[3, 5, 7, 9]$
Decoder kernel size inc	categorical	$[3, 5, 7, 9]$
Number of convolutional layers	integer	$[1, 5]$
Dropout decoder	float	$[0.05, 0.50]$, step 0.05
Batch size	categorical	$[16, 32]$
Learning rate	log-uniform	$[3 \times 10^{-5}, 3 \times 10^{-3}]$
Weight decay	log-uniform	$[1 \times 10^{-6}, 1 \times 10^{-2}]$
Percentage start	float	$[0.20, 0.40]$, step 0.05

4.2 Performance Metrics

The evaluation of the performance considers both the ability of the model to detect maneuver events and its ability to accurately reconstruct the corresponding maneuver intervals.
A maneuver is considered successfully detected if the model correctly identifies at least one sample belonging to the corresponding maneuver interval. The model output consists of a set of overlapping windows, each containing 512 prediction scores. A prediction threshold of 0.9 is adopted to classify individual samples: values greater than 0.9 are labeled as maneuver samples, while lower values are classified as non-maneuver samples. Since each sample may appear in multiple overlapping windows, a voting procedure is applied to obtain a single prediction for each global time index. Specifically, for each sample, the number of times it is classified as a maneuver is divided by the total number of windows in which it appears. If this ratio exceeds 50%, the corresponding global sample is labeled as a maneuver; otherwise, it is classified as non-maneuver. This procedure produces a prediction sequence with the same length as the ground-truth mask, enabling a direct comparison between predictions and reference labels. To evaluate maneuver detection, a predicted maneuver is considered a true positive (TP) if it overlaps with at least one sample belonging to a ground-truth maneuver interval. Predicted maneuver intervals that do not overlap with any ground-truth maneuver are counted as false positives (FP), while ground-truth maneuver intervals with no corresponding prediction are counted as false negatives (FN). The detection performance is evaluated using the following metrics:

$$\text{Precision(P)} = \frac{TP}{TP+FP} \tag{7}$$

$$\text{Recall(R)} = \frac{TP}{TP+FN} \tag{8}$$

$$\text{F1} = 2 \times \frac{P \times R}{P+R} \tag{9}$$

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Maneuver reconstruction refers to the ability of the model not only to detect the occurrence of a maneuver, but also to accurately identify its entire temporal extent. In other words, the predicted maneuver interval should closely match the corresponding ground-truth interval. Since the ground-truth maneuver masks were manually refined, slight inaccuracies in the interval boundaries may be present. To account for this uncertainty, the quality of the reconstructed maneuver intervals is evaluated using the Intersection over Union (IoU) metric, which is widely adopted in object detection:

$$\mathrm{IoU} = \frac{\left| I_{\mathrm{pred}} \cap I_{\mathrm{gt}} \right|}{\left| I_{\mathrm{pred}} \cup I_{\mathrm{gt}} \right|} \tag{10}$$

where I_{pred} and I_{gt} denote the predicted and ground-truth maneuver intervals, respectively.

A maneuver reconstruction is considered correct if its IoU exceeds 0.5. This threshold avoids penalizing accurate reconstructions that may differ slightly from the manually defined ground-truth intervals. Reconstruction performance is evaluated as the percentage of correctly reconstructed maneuvers with respect to the total number of maneuvers detected by the model.

4.3 Training and Testing Pipeline

The performance of the proposed method is evaluated from two complementary perspectives. First, the model is tested on the test set, which consists of orbital parameter segments extracted from satellites also included in the training set. This evaluation measures the ability of the model to detect maneuvers on unseen portions of trajectories belonging to known satellites. Furthermore, the model is evaluated on satellites that were excluded from the training phase. This experiment is used to test the generalization capability of the model and to demonstrate that it can maintain comparable performance when applied to previously unseen satellites.

Table 4 reports the results obtained on the test set. The table includes the total number of maneuvers, the number of true positives, false positives, and false negatives for maneuver detection, together with the corresponding F_1-score. Regarding maneuver reconstruction, the table reports the number of true positives, corresponding to the number of maneuvers with an IoU value greater than 0.5 and therefore considered correctly reconstructed, as well as the percentage of correctly reconstructed maneuvers with respect to the total number of detected maneuvers.

Table 4: Performance evaluated on the test set for both the semi-major axis and inclination parameters.

Orbital Parameter	#Maneuver	TP	$\mathrm{TP}_{reconstruction}$	FP	FN	F_1-score	$\%_{reconstruction}$
Semi-major axis	111	106	98	3	5	0.964	92.4
Inclination	8	8	8	0	0	1	100

Table 5 and Table 6 report the results obtained using the same evaluation metrics presented in Table 4 for satellites that were not included in the training set. To demonstrate the generalization capability of the proposed model, the objective is to achieve comparable performance on unseen satellites with respect to those used during training. Table 5 presents the results obtained for the semi-major axis, whereas Table 6 reports the corresponding results for the inclination parameter. These results are comparable to those reported in Table 4, demonstrating the ability of the proposed model to generalize maneuver detection across different satellites. For both evaluation scenarios, the maneuver detection performance reaches an F_1-score of approximately 0.95, while more than 90% of the detected maneuvers are correctly reconstructed. An additional aspect is that several false positives correspond to isolated samples identified as maneuvers in close proximity to the real maneuver intervals. This behavior may be attributed to slight inaccuracies in the manually refined maneuver boundaries used to generate the ground-truth labels, particularly for complex maneuver profiles such as those of SPOT-2, which exhibits the highest number of false positives. Consequently, many of these false positives can be mitigated through a post-processing stage, further reducing their overall number. Regarding the false negatives, which cannot be recovered through post-processing, a significant fraction corresponds to maneuvers that decrease the semi-major axis. This type of maneuver is underrepresented in the training dataset, as most LEO satellites predominantly perform maneuvers that increase the value of the semi-major axis to counteract the effect of atmospheric drag. As a result, the model has fewer examples from which to learn this maneuver pattern, leading to a slightly lower detection performance for such events.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Table 5: Performance evaluated on satellites not included in the training dataset for the semi-major axis parameter.

Satellite	#Maneuver	TP	$TP_{reconstruction}$	FP	FN	F_1-score	$\%_{reconstruction}$
Jason-3	44	42	39	5	2	0.923	92.9
Sentinel-3B	73	69	69	1	4	0.965	100
Sentinel-6A	21	21	18	1	0	0.977	85.7
SPOT-2	107	103	99	11	4	0.932	96.1
SWOT	38	37	37	0	1	0.987	100
Haiyang-2A	31	30	26	3	1	0.937	86.7
Overall	314	302	288	21	12	0.948	95.4

Table 6: Performance evaluated on satellites not included in the training dataset for the inclination parameter.

Satellite	#Maneuver	TP	$TP_{reconstruction}$	FP	FN	F_1-score	$\%_{reconstruction}$
Jason-3	0	0	0	0	0	-	-
Sentinel-3B	18	18	17	0	0	1	94.4
Sentinel-6A	0	0	0	0	0	-	-
SPOT-2	10	10	9	1	0	0.952	90
SWOT	0	0	0	0	0	-	-
Haiyang-2A	0	0	0	0	0	-	-
Overall	28	28	26	1	0	0.982	92.9

Fig. 4 shows a portion of the Haiyang-2A orbital parameters spanning the period from 2014-01-12 to 2016-02-26. This segment is particularly challenging because the semi-major axis time series is characterized by a high level of noise, with significant oscillations and numerous spikes. Despite these conditions, the proposed model accurately identifies the maneuver events, missing only one maneuver while generating two false positives. The only metric partially affected by the increased noise is the maneuver reconstruction rate, which decreases slightly to below 90%. This behavior is due to the high noise level, which makes both maneuver detection and the manual generation of accurate ground-truth labels more challenging than for the other satellites considered in this study.

An example of the output produced by the training and testing pipeline is shown in Fig. 5. The figure presents the results obtained for Sentinel-3B, a satellite that was not included in the training set. It compares the maneuver intervals predicted by the proposed model with the corresponding ground-truth maneuvers, highlighting the ability of the model to accurately reconstruct the maneuver history of an unseen satellite.

4.4 Operational Pipeline

As stated previously, the objective of this work is to develop a model that can be applied to satellites for which no maneuver history is available, enabling the reconstruction of such information from publicly available data. This subsection presents the results obtained using the proposed operational pipeline. In this setting, the trained model receives as input only the pre-processed time series of semi-major axis and inclination, without any ground-truth maneuver labels.

Unlike the evaluation phase, the network output is not used to compute performance metrics. Instead, it is directly exploited to reconstruct the maneuver history of the analyzed satellite, with the aim of generating a reference dataset that can be used as ground truth for future applications. The final output of the proposed pipeline is a CSV file containing six columns, organized as shown in Table 7. The table presents a part of the output generated for Sentinel-2B, reporting the first four maneuvers detected on the semi-major axis. The *parameter* column identifies the orbital parameter on which the maneuver has been detected. The *start idx* and *end idx* columns correspond to the *pre-maneuver TLE* and *post-maneuver TLE*, respectively, as defined in Section 3.2. These indices delimit the maneuver interval by identifying the first and last TLE associated with the detected maneuver. The *start utc* and *end utc* columns report the corresponding UTC timestamps of the maneuver interval. Finally, the *n tle* column indicates the number of TLEs included in the detected maneuver interval.

This subsection presents qualitative results for two representative satellites, whose characteristics are summarized in

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

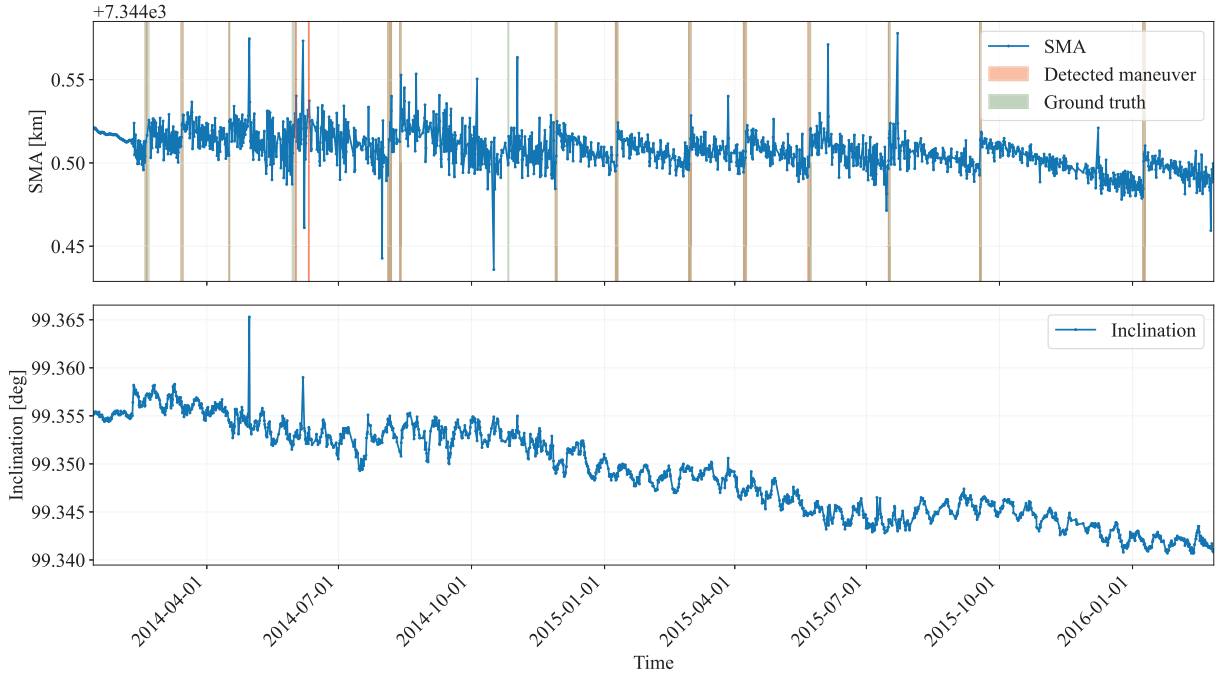


Fig. 4: Maneuver history of Haiyang-2A from 2014-01-12 to 2016-02-26.

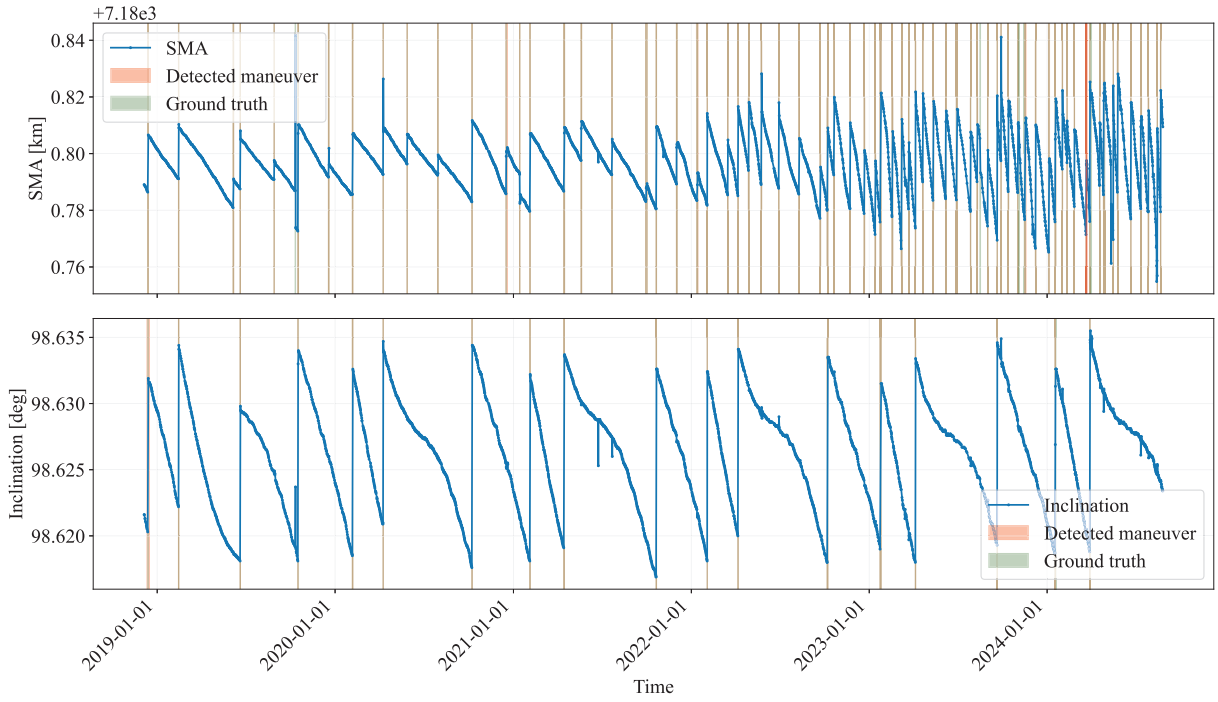


Fig. 5: Maneuver history of Sentinel-3B.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Table 7: Example of the CSV file generated by the operational pipeline.

parameter	start idx	end idx	start utc	end utc	n tle
a	47	62	2017-05-10T02:27:41	2017-05-16T17:54:09	16
a	143	148	2017-06-29T20:55:27	2017-07-04T03:18:07	6
a	453	459	2017-09-27T19:14:31	2017-09-29T19:54:47	7
a	466	486	2017-10-01T12:11:33	2017-10-07T17:33:49	21

Table 8. Fig. 6 shows the results obtained for Landsat 9, while Fig. 7 presents the corresponding results for Sentinel-2B.

Table 8: Satellite Dataset for Operational Pipeline.

Satellite	Start Date	End Date	Altitude (km)	Inclination (deg)
Landsat 9	2021-12-06	2026-06-10	~ 705	~ 98.2
Sentinel-2B	2017-04-02	2026-06-09	~ 780	~ 98.56

A visual inspection of Fig. 6 and Fig. 7 shows that every change in the semi-major axis attributable to a maneuver is matched by a detected interval of comparable extent, consistent with the detection and reconstruction behavior observed on the labeled satellites. A more detailed example of the reconstruction capability is provided in Fig. 8, where the model accurately reconstructs the entire variation in the semi-major axis time series due to the maneuver. On the inclination parameter, not all changes are matched by a detected interval, particularly for Sentinel-2B, as shown in Fig. 7. This behavior can be attributed to the limited number of inclination maneuvers available in the training set. In particular, most inclination maneuvers in the training data span only two TLEs, and the model correctly detects this type of maneuver for both Landsat 9 and Sentinel-2B. The inclination changes that are not matched by a detected interval are mostly those extending over multiple TLEs, which are underrepresented in the training dataset. As a result, the reconstruction capability of the model is also reduced for these more challenging maneuver profiles, as shown in Fig. 8.

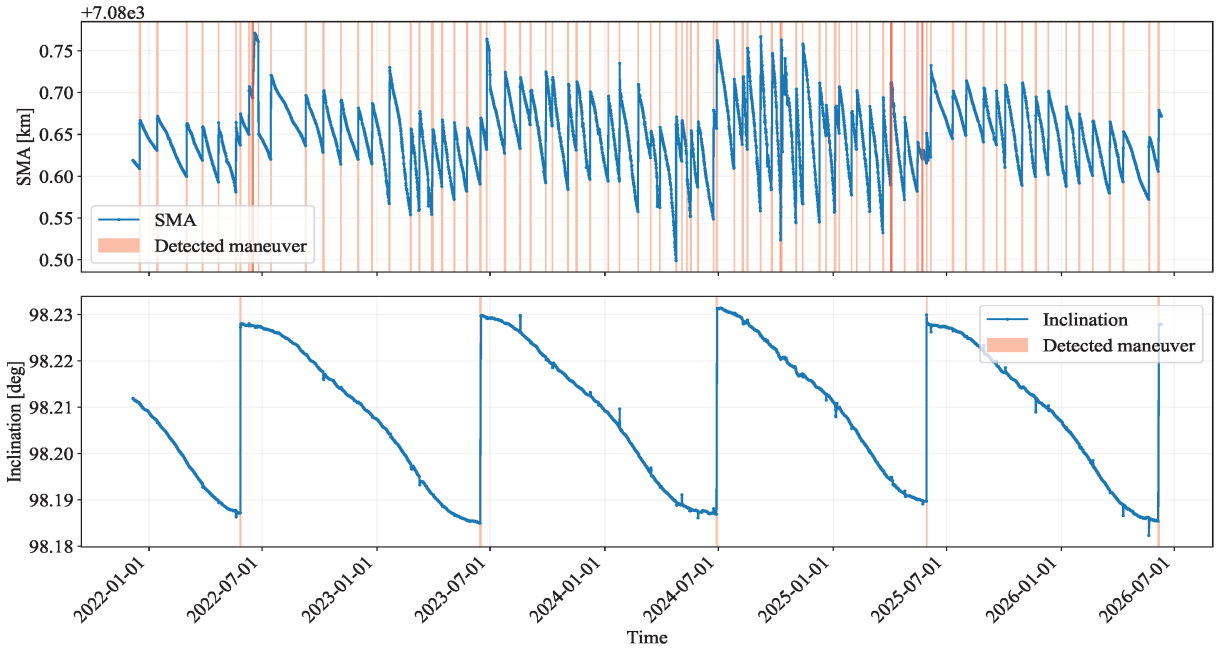


Fig. 6: Maneuver history of Landsat 9.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

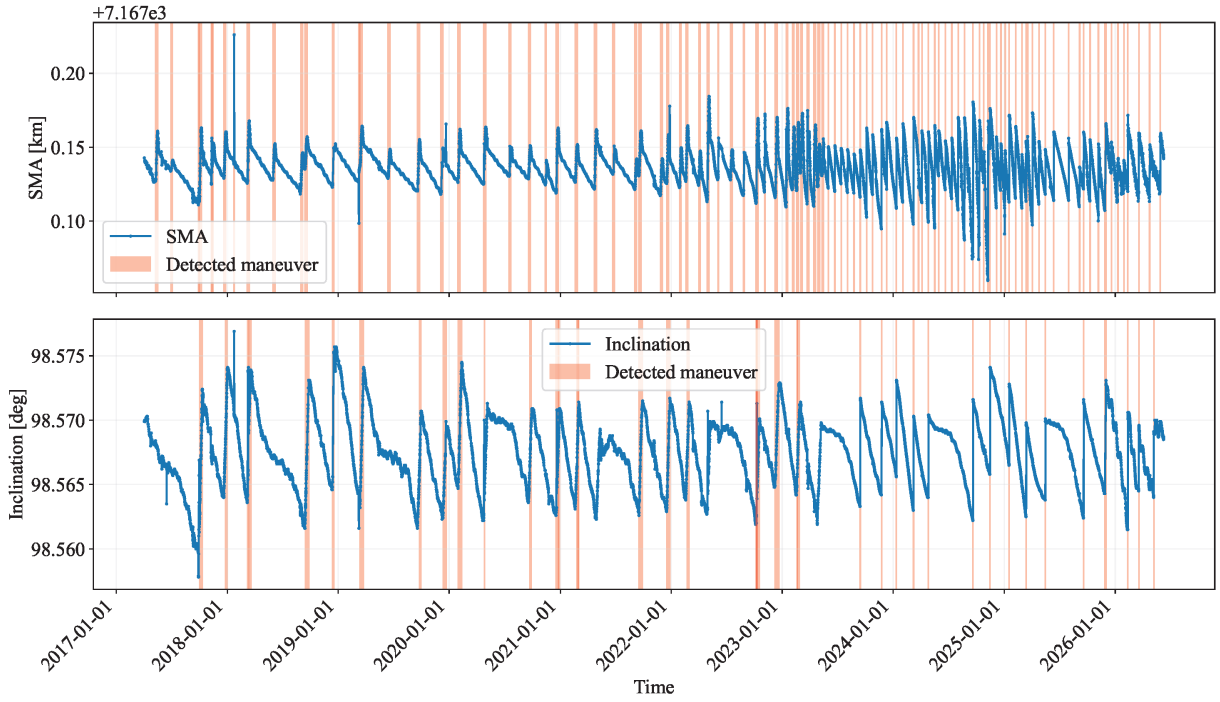


Fig. 7: Maneuver history of Sentinel-2B.

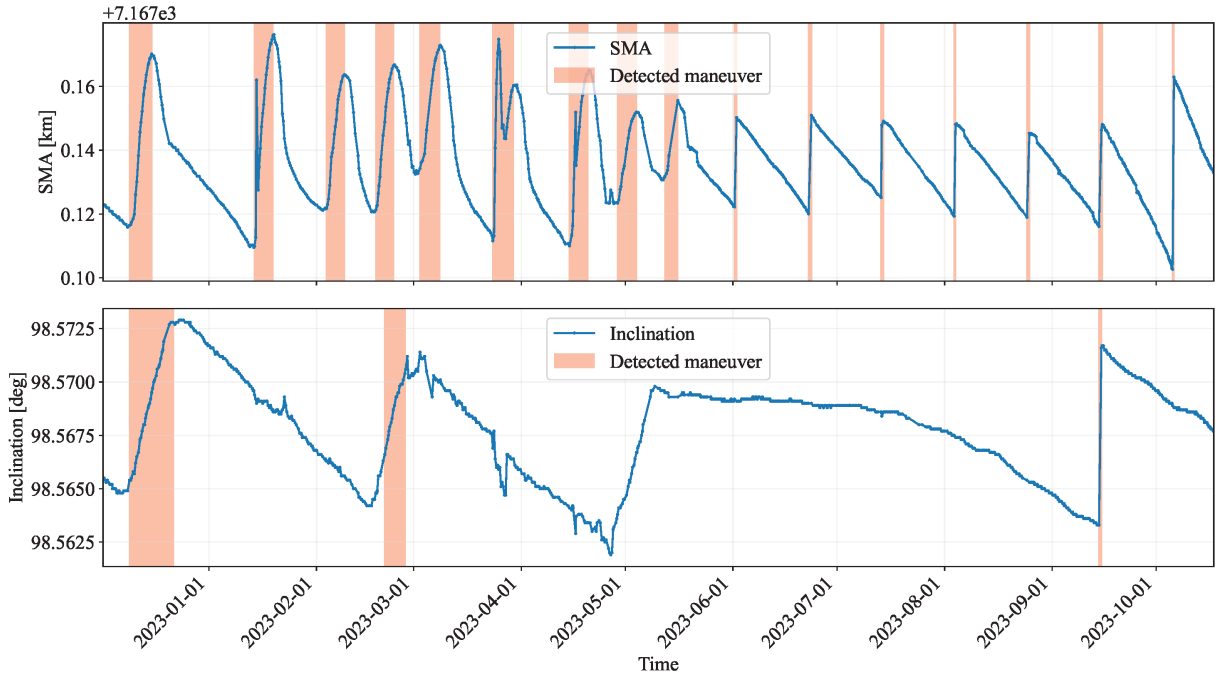


Fig. 8: Maneuver history of Sentinel-2B from 2022-12-01 to 2023-10-17.

4.5 Comparison with Classical Method

In this subsection, the performance of the proposed approach is compared with a classical maneuver detection method that does not rely on deep learning techniques. Specifically, the method proposed by Patera [3] is considered, as it

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

represents one of the reference statistical approaches for maneuver detection. The main limitation of this method is that the thresholds, window sizes, and the order of the polynomial used to fit the data are parameters that need to be tuned for each satellite. This requirement significantly limits the scalability of the method in operational scenarios, where the maneuver history of the target satellite is unknown. The purpose of this comparison is to highlight that, although the method proposed by Patera can achieve good performance when its hyperparameters are optimized for a specific satellite, its performance degrades when the same configuration is applied to different satellites. In contrast, the proposed approach applies the same trained network across different satellites while maintaining consistently high and comparable performance. The comparison is carried out using the Sentinel-3B and SPOT-2 satellites. The hyperparameters of the method of Patera were optimized through a grid search procedure, with the objective of maximizing the F_1-score for both the semi-major axis and inclination on Sentinel-3B. The resulting performance on Sentinel-3B is reported in Table 9 and Table 10 and compared with that of the proposed approach. It is important to highlight that the results obtained by the proposed method are achieved without training the model on Sentinel-3B, demonstrating its ability to generalize to previously unseen satellites.

Table 9: Performance evaluated on the semi-major axis of Sentinel-3B.

Method	#Maneuver	TP	FP	FN	F_1-score
Patera	73	67	3	6	0.937
Proposed Approach	73	69	1	4	0.965

Table 10: Performance evaluated on the inclination of Sentinel-3B.

Method	#Maneuver	TP	FP	FN	F_1-score
Patera	18	18	0	0	1
Proposed Approach	18	18	0	0	1

The hyperparameters optimized on Sentinel-3B were subsequently used without modification to apply the method of Patera to SPOT-2. The corresponding results are reported in Table 11 and Table 12.

Table 11: Performance evaluated on the semi-major axis of SPOT-2.

Method	#Maneuver	TP	FP	FN	F_1-score
Patera	107	76	27	31	0.724
Proposed Approach	107	103	11	4	0.932

Table 12: Performance evaluated on the inclination of SPOT-2.

Method	#Maneuver	TP	FP	FN	F_1-score
Patera	10	9	1	1	0.9
Proposed Approach	10	10	1	0	0.952

These results indicate that the method of Patera is highly dependent on the tuning of the hyperparameters discussed previously, particularly the detection threshold used to identify maneuver events. The degradation of the performance is especially significant for the semi-major axis. This can be explained by the different characteristics of the maneuver profiles: the inclination maneuvers of SPOT-2 are similar to those of Sentinel-3B, whereas many semi-major axis maneuvers span more than two TLEs, producing more gradual changes in the orbital parameter. In contrast, the semi-major axis maneuvers of Sentinel-3B are mainly characterized by abrupt jumps involving only two TLEs. Consequently, the hyperparameters optimized for Sentinel-3B cannot be directly applied to SPOT-2, requiring a dedicated tuning process for each satellite. This limitation reduces the scalability of the method of Patera for operational applications involving satellites with unknown maneuver histories.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

5. CONCLUSIONS

This work presented a data-driven framework for satellite maneuver detection based exclusively on publicly available TLE data. The primary objective was to develop a model capable of generalizing across different satellites, allowing maneuver detection to be performed on satellites not observed during training while maintaining performance comparable to that achieved on satellites present in the training set. A second objective was to reconstruct the temporal extent of each maneuver, rather than simply detecting its occurrence, with the ultimate goal of reconstructing the maneuver history of satellites for which this information is not publicly available. To achieve these objectives, a neural architecture composed of a Transformer-based encoder and a convolutional decoder is proposed. The semi-major axis and inclination time series are processed through two independent streams, allowing the model to learn temporal patterns of the parameters. The experimental results demonstrate that the proposed approach achieves strong performance in both maneuver detection and reconstruction, showing that the proposed model generalizes well to previously unseen satellites. The main limitations of the proposed approach are related to the limited availability of publicly accessible maneuver data and to the manual refinement of the maneuver labels. The scarcity of publicly available maneuver histories results in some maneuver types being underrepresented in the training dataset, such as inclination maneuvers spanning multiple TLEs. In addition, although manually refining the maneuver intervals improves the quality of the ground truth, slight inaccuracies at the maneuver boundaries can still remain and affect the reconstruction performance. Finally, the model remains reliable on highly noisy TLE time series, but noise both increases the number of false positives and degrades the reconstructed maneuver intervals, as it blurs the maneuver boundaries for the network and for the manual labeling. A promising direction for future work is to address the limited availability of maneuver data by applying the proposed framework to satellites whose maneuver histories are unknown. After a visual inspection and possible refinement of the reconstructed maneuver intervals, the reconstructed maneuver labels could be combined with the existing ILRS ground truth to build a larger and more representative maneuver dataset, enabling further improvements in model generalization. Another possible extension is the introduction of a cross-attention module between the two input streams. This would allow the model to capture dependencies not only along each orbital parameter, but also between the semi-major axis and inclination. Since changes in these parameters often occur simultaneously during orbital maneuvers, exploiting their correlation could further improve the detection and reconstruction of more complex maneuver profiles. A further extension of this work could focus on the reconstruction of additional maneuver characteristics, such as the associated ΔV. This would represent a first step toward a more complete maneuver characterization, including the identification of different maneuver types. Overall, the proposed framework represents a promising step toward scalable machine learning approaches for satellite maneuver detection, enabling the reconstruction of maneuver histories using only publicly available TLE data.

6. REFERENCES

[1] European Space Agency. ESA's annual space environment report. Technical report, Space Debris Office, 2025.

[2] Tom Kelecy, Doyle Hall, Kris Hamada, and Dennis Stocker. Satellite maneuver detection using two-line element (TLE) data. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, pages 1–10. Maui Economic Development Board (MEDB) Maui, HI, 2007.

[3] Russell P Patera. Space event detection method. *Journal of Spacecraft and Rockets*, 45(3):554–559, 2008.

[4] Stijn Lemmens and Holger Krag. Two-line-elements-based maneuver detection methods for satellites in low earth orbit. *Journal of Guidance, Control, and Dynamics*, 37(3):860–868, 2014.

[5] Xue Bai, Chuan Liao, Xiao Pan, and Ming Xu. Mining two-line element data to detect orbital maneuver for satellite. *IEEE Access*, 7:129537–129550, 2019.

[6] Rohit Mital, Kim Cates, Joe Coughlin, and Geetha Ganji. A machine learning approach to modeling satellite behavior. In *2019 IEEE international conference on space mission challenges for information technology (SMC-IT)*, pages 62–69. IEEE, 2019.

[7] Phil DiBona, James Foster, Anthony Falcone, and Michael Czajkowski. Machine learning for RSO maneuver classification and orbital pattern prediction. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, number 54, page 54. Maui Economic Development Board Kihei, Hawaii, 2019.

[8] Thomas G Roberts and Richard Linares. Geosynchronous satellite maneuver classification via supervised machine learning. In *Advanced Maui Optical and Space Surveillance Technologies Conference. Maui, Hawaii*, 2021.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

[9] Riccardo Cipollone, Italo Leonzio, Gaetano Calabrò, and Pierluigi Di Lizia. An LSTM-based maneuver detection algorithm from satellites pattern of life. In *2023 IEEE 10th international workshop on metrology for aerospace (Metroaerospace)*, pages 78–83. IEEE, 2023.

[10] Riccardo Cipollone, Carolin Frueh, and Pierluigi Di Lizia. LSTM-based maneuver detection for resident space object catalog maintenance. *Neural Computing and Applications*, 37(22):17341–17362, 2025.

[11] R Cipollone, E Raviola, and P Di Lizia. An unsupervised learning-based manoeuvre detection method for resident space object pattern of life characterisation. In *9th European Conference on Space Debris, Bonn, Germany*, 2025.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6778–6786, 2023. Survey Track.

[14] Peng Mun Siew, Haley E Solera, Giovanni Lavezzi, Thomas G Roberts, Daniel Jang, David Baldsiefen, B Tran, C Yeung, K Johnson, N Metzger, et al. Satellite pattern-of-life identification challenge: competition design and results. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, page 135, 2024.

[15] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[16] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[18] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Copyright © 2026 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com