Deep Matrix Profile for Maneuver Classification in Low Earth Orbit Satellite Trajectories

Stéfan Baudier, Santiago Velasco-Forero, Franck Jean, Daniel Brooks, Jesus Angulo

Deep Matrix Profile for Maneuver Classification in Low Earth Orbit Satellite Trajectories

Stéfan Baudier*†, Santiago Velasco-Forero*, Franck Jean†, Daniel Brooks† and Jesus Angulo*
Mines Paris, PSL University *
Centre for Mathematical Morphology (CMM)
Fontainebleau, France
stefan.baudier@minesparis.psl.eu
Thales LAS France †
Advanced Radar Concepts (ARC)
Limours, France

***Abstract*—The characterization of satellite behavior is of paramount importance in Space Surveillance Awareness. One crucial aspect of studying satellite behavior involves the detection of out-of-station-keeping maneuvers within satellite trajectories. Since each satellite's station-keeping behavior is unique, extracting relevant information from historical data is crucial for modeling the Pattern of Life. In this context, we propose integrating a state-of-the-art data mining technique, namely the Matrix Profile, into a neural network architecture. Our approach highlights the connection with non-local models. The proposed model exhibits promising results for detecting out-of-station-keeping maneuvers in Low Earth Orbit satellite trajectories, showcasing its effectiveness in addressing the challenges of unbalanced data.**

***Index Terms*—maneuver classification, satellite trajectories, station keeping, matrix profile.**

I. Introduction

In recent years, private businesses have been interested in exploiting space, primarily through Low Earth Orbit (LEO) satellite usage. Considering the space growth policy, it will continue to increase in the following years. As the near space becomes more populated, issues such as space debris collisions or geopolitical conflicts appear. The increase in space assets leads to an explosion of related data. However, their exploitation in the space surveillance awareness domain mainly rests on human intervention, which is time-consuming and error-prone. It raises the need for automated treatment for higher reactivity and more precise models. This work aims to support these changes by detecting anomalies in the repeatable and predictable behavior of the satellite called the Pattern of Life (PoL).

Once in orbit, satellites drift from their original trajectories due to physical perturbations like the atmospheric drag [1]. In this case, the satellite loses altitude and needs to compensate for this fall by performing station-keeping maneuvers. Satellites can also use maneuvers for different purposes, such as changing their mission or spying. Standard methods to detect out-of-PoL behaviors treat the out-of-station-keeping maneuver detection. However, most of this work deals with GEostationary Orbits (GEO) satellites or Two-Line Elements (TLE) as input, for example [2], [3]. Here, considering the PoL as natural trajectories and station-keeping maneuvers, we aim to classify whether a trajectory belongs to the PoL, focusing on the case of LEO satellites using ephemeris data. There is no reason for the station-keeping process to be similar from one satellite to another. Then, we propose representing the satellite station-keeping process by its individual historical behavior. Our approach involves employing supervised machine learning techniques tailored for time series analysis. As maneuvers are particular patterns of several burns, this issue corresponds to the detection of collective and contextual anomalies defined in [4], [5]. Matrix Profile (MP) [6] is a state-of-the-art tool in data mining for discord computation, motif detection, and anomaly detection. It computes the distances between each window and its nearest neighbors in a time series. This is in the scope of collective and contextual anomalies. It has already been used with neural networks for data pre-processing and feature extraction [7], [8]. Another approach, including MP computation in neural networks, has been recently published [9].

Additionally, we highlight the relationship between the matrix profile and similarity computation in neural networks through attention layers [10], as well as transformers, which represent state-of-the-art methods for time series analysis [11]. The Autoformer [12] utilizes correlation computation directly as an attention layer, indicating a strong association with the matrix profile due to the ability to compute correlation coefficients instead of distances [13]. Given that attention layers exemplify non-local neural networks [14], [15], one might question whether a matrix profile can be viewed as a non-local function within a neural network.

The contributions of our study are summarized as follows. i) We unveil the relationship between the Matrix Profile and Nonlocal models. ii) We propose two new neural network architectures inspired by the matrix profile algorithm called Deep Matrix Profile (DMP). iii) These models are applied explicitly to maneuver classification in LEO satellite trajectories.

II. Matrix profile as Non-local operator

A matrix profile [6] is a distance vector between each window and its nearest neighbors in a time series. The Scalable Time series Anytime Matrix Profile (STAMP) algorithm is one of the fundamental algorithms that approximates the matrix profile. It mainly comprises the MASS (Mueen's Algorithm for Similarity Search) [16] algorithm that allows a fast computation of the distance profiles used to compute the matrix profile. We propose the integration of matrix profile

computation within neural networks as a non-local function. In doing so, we elucidate the core principles underlying the MASS and STAMP algorithms, demonstrate the alignment of matrix profile computation with the non-local networks paradigm, and introduce two novel non-local functions.

A. MASS algorithm

The MASS algorithm computes a *distance profile*, which is the vector of distances between a given time series $y = [y_1, \cdots, y_m] \in \mathbb{R}^m$ called the *query* and every sub-window of length m of s, with $s \in \mathbb{R}^n$ and $m \leq n$.

1) Distance computation between two sub-windows: Let $s^{(i)} := [s_{i+1}, \cdots, s_{i+m}] \in \mathbb{R}^m$ a sub-window of $s \in \mathbb{R}^n$. We consider that $\hat{y}$ is the z-normalized vector of y *i.e.* $\hat{y} := \frac{y-\mu_y}{\sigma_y}$ and the z-distance d_z is the Euclidean distance that includes z-normalization for both vectors,

$$d_z^2(s^{(i)}, y) := \sum_{j=1}^{m} \left(\frac{s_{i+j} - \mu_{s^{(i)}}}{\sigma_{s^{(i)}}} - \frac{y_j - \mu_y}{\sigma_y} \right)^2. \quad (1)$$

In a few simple steps, one obtains from (1):

$$d_z^2(s^{(i)}, y) = 2m \left(1 - \frac{1}{m} \sum_{j=1}^{m} \left(\frac{s_{i+j} - \mu_{s^{(i)}}}{\sigma_{s^{(i)}}} \right) \left(\frac{y_j - \mu_y}{\sigma_y} \right) \right) \quad (2)$$

From now on, one will suppose that y is already z-normalized *i.e.* $\mu_y = 0$ and $\sigma_y = 1$. Then, we obtain from (2):

$$d_z(s^{(i)}, y) = \sqrt{2 \left(m - \frac{\sum_{j=1}^{m} s_{i+j} y_j}{\sigma_{s^{(i)}}} \right)}. \quad (3)$$

2) Efficient computation of d_z components: Let us first compute the $\sigma_{s^{(i)}}$ terms in (3) *i.e.* $\Theta_s := [\sigma_{s^{(0)}}, \cdots, \sigma_{s^{(n-m)}}]$ using cumulative sums:

$$\sigma_{s^{(i)}} = \sqrt{\frac{C^2_{i+m} - C^2_i}{m} - \left(\frac{C_{i+m} - C_i}{m} \right)^2} \quad (4)$$

with $C_i := \sum_{j=1}^{i} s_j$ and $C_i^2 := \sum_{j=1}^{i} s_j^2$. Considering $C_{k,l} := [C_k, \cdots, C_l]$ and $C^2_{k,l} := [C^2_k, \cdots, C^2_l]$, we obtain:

$$\Theta_s = \sqrt{\frac{C^2_{m,n} - C^2_{1,n-m}}{m} - \left(\frac{C_{m,n} - C_{1,n-m}}{m} \right)^2}. \quad (5)$$

On the other hand, one can compute the dot product term in (3) via a convolution. Let us denote the dot product by $dp_i := \sum_{j=1}^{m} s_{i+j} y_j$, the vector varying the index i by $DP_{s,y} := [dp_1, \cdots, dp_{n-m}]$ and the reversed padded query $y^* := [y_m, \cdots, y_1, 0, \cdots, 0] \in \mathbb{R}^n$. By defining the convolution,

$$\mathit{conv}(s, y^*)[i] := \sum_{j=1}^{i} s_j y^*_{i-j+1} \quad (6)$$

and considering that a part of y^* is zeros and modifying the index, we obtain

$$\mathit{conv}(s, y^*)[i] = \sum_{j=i-m+1}^{i} s_j y^*_{i-j+1} = \sum_{j=1}^{m} s_{j+i-m} y_j. \quad (7)$$

One observes that

$$\mathit{conv}(s, y^*)[i+m] = \sum_{j=1}^{m} s_{j+i} y_j = dp_i, \quad (8)$$

then the dot product term can be computed via convolution by

$$DP_{s,y} = \mathit{conv}(s, y^*). \quad (9)$$

The convolution computation using a FFT reduces the complexity from $\mathcal{O}(nm)$ to $\mathcal{O}(n \log(n))$ compared with the classic convolution. We use the classical convolution operators because they are better optimized in deep learning frameworks using GPUs. The formalism uses only convolutions and allows them to be used as a deep learning layer.

3) Distance profile computation: Finally, considering an already z-normalized query y, one can compute the distance profile of y over s with

$$D(s, y) := \sqrt{2(m - \frac{DP_{s,y}}{\Theta_s})}. \quad (10)$$

Note that the term $\frac{DP_{s,y}}{\Theta_s}$ is equal to m times the correlation between s and y [13] wich is the recommendation for similarity in the Autoformer models [12].

B. Matrix Profile

The matrix profile [6] consists of computing the distance between each sub-window of a time series s^A and their nearest neighbor in s^B by computing distance profiles in (10). We utilize two methods to conduct this search:

1) Greedy algorithm: The greedy way of computing a matrix profile is to iterate on all the subwindows of s^A, compute their distance profile over s^B, and then keep the minimum for each distance profile.

2) Scalable Time series Anytime Matrix Profile (STAMP): The STAMP algorithm [6] iterates on the subwindows of s^B and computes their distance profile over s^A. The goal here is not to compute one precise component of the matrix profile at each iteration but to refine the matrix profile values. The pairwise minimum between the distance profile and the previous matrix profile is computed at each iteration. With a random walk over the s^B sub windows, the STAMP converges quickly to the exact values of the matrix profile. It enables iterating over a fraction of the queries to obtain a good approximation of the matrix profile (empirically 10%). It leads to a high computational time saving, especially when the length of s^A and/or s^B are high.

C. Connection to non-local networks

One wants to detect a deviant collection of values compared to past collections. Let $\mathbf{S} := [\mathbf{S}_1, \cdots, \mathbf{S}_n] \in \mathbb{R}^{\mathrm{d} \times n}$, be a multivariate time series, and $\mathbf{S}^{(i)} := [\mathbf{S}_{i+1}, \cdots, \mathbf{S}_{i+m}]$ be a window to classify in $\mathbf{S}$, in our case the satellite trajectory. Let $\mathbf{S}^{\leq i} := [\mathbf{S}_1, \cdots, \mathbf{S}_i]$ be the *historical context* of $\mathbf{S}^{(i)}$, describing the past behavior of the satellite. We want to classify the window $\mathbf{S}^{(i)}$ as having normal or abnormal behavior compared to $\mathbf{S}^{\leq i}$.

Non-local neural networks are defined in [14] [15] as networks including non-local operations over data support like images, videos, or time series. These operations aggregate in one position weighted information from all the positions. Formally,

$$o_j := \frac{1}{C} \sum_{i=1}^{n} f(x_i, x_j) g(x_i) \tag{11}$$

With C the normalization factor, $[x_1, \cdots, x_n]$ all positions of an input signal x and o the output signal with the same size, where f and g are parametrized function learned from data. One proposes a slight modification of this paradigm. First, the non-local information could come from another support, i.e., o and x have different sizes. Secondly, in the time series case, one supposes that the x_i and x_j could be sub-windows of time series. It leads to:

$$o_j = \frac{1}{C} \sum_{i=1}^{a-m+1} f(\mathbf{S}_i^A, \mathbf{S}_j^B) g(\mathbf{S}_i^A), \tag{12}$$

with $[\mathbf{S}_1^A, \cdots, \mathbf{S}_{a-m+1}^A]$ all sub-windows of $\mathbf{S}^A$ and $[\mathbf{S}_1^B, \cdots, \mathbf{S}_{b-m+1}^B]$ all sub-windows of $\mathbf{S}^B$, where $a := |\mathbf{S}^A|$, $b := |\mathbf{S}^B|$ and m the sub-windows size.

Let $\texttt{amin}$ be the indicator function,

$$\texttt{amin}(x_i) := \begin{cases} 1, & \text{if } x_i = \min(x) \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

We introduce a novel layer for deep learning models based on distance profile computation (10) and the non-local formalism in (12):

$$o_j = \sum_{i=1}^{a-m+1} \underset{j}{\texttt{amin}}(d_z(f(\mathbf{S}_i^A), f(\mathbf{S}_j^B))) d_z(f(\mathbf{S}_i^A), f(\mathbf{S}_j^B)) g(\mathbf{S}_i^A). \tag{14}$$

Note that for g equal to one, (14) corresponds to one component of a Matrix Profile in II-B. Finally, we propose a soft version of the matrix profile that takes into account all the contributions but weighs them according to their respective Euclidean distance values,

$$o_j = \sum_{i=1}^{a-m+1} \underset{j}{\texttt{smin}}(d_z(f(\mathbf{S}_i^A), f(\mathbf{S}_j^B))) d_z(f(\mathbf{S}_i^A), f(\mathbf{S}_j^B)) g(\mathbf{S}_i^A). \tag{15}$$

where $\texttt{smin}(x_i) := \frac{\exp(-x_i)}{\sum_j \exp(-x_j)}$ is a smooth approximation to the minimum function.

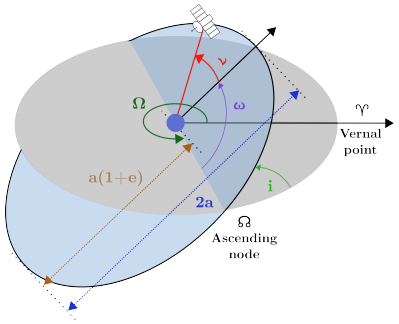


Fig. 1: Representation of Keplerian parameters: a is the semi-major axis, i the inclination, e the eccentricity, ω the argument of periapsis, Ω the longitude of the ascending node, and ν the true anomaly.

III. Proposed Method

A. Data

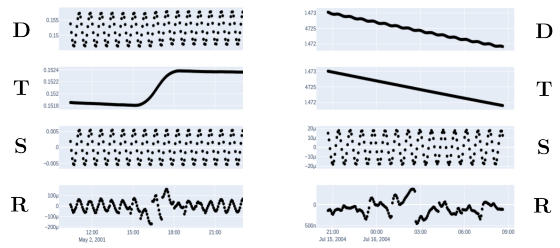
We use simulated data of satellite trajectories based on high-fidelity physics models and real satellite characteristics [17]. These trajectories contain station-keeping maneuvers, triggered via a servo system, that aim to correct the orbits' semi-major axis and the inclination Fig. 1. They respectively describe the size of the orbit and its orientation compared with the Earth's equatorial plane. Additionally, these scenarios contain two distinct types of maneuvers that deviate from this typical behavior. These anomalous maneuvers constitute the class we aim to detect in our experiments. The first type resembles station-keeping maneuvers in intensity, modifying exclusively the semi-major axis or the inclination. However, they do not adhere to the servo system strategy. On the other hand, the second type exhibits even more significant deviations in intensity compared to the station-keeping distribution. It does not target any specific aspect of the trajectory for alteration. All these maneuvers are composed of impulsive burns.

These data comprise 400 LEO satellite scenarios (one-year trajectory), 250 with random trajectories, and 150 with Starlink look-alike trajectories. One notes that each scenario is based on variable inner parameters, notably making the station-keeping strategy unique for each scenario.

We use ephemeris data as input for our model. They describe the position and velocity of the satellite with a regular timestamp. They are periodic with several periodicities. The osculating orbits represent the Keplerian orbit Fig. 1 that the satellite should have at every timestamp if there were no perturbations, e.g., only two-body problems with isotropic objects. We use the equinoctial parameters, a variant of Keplerian orbit parameters, that do not suffer from singularities [18].

B. Data preprocessing

We used the Seasonal-Trend decomposition procedure based on Loess (STL) [19], a filter that decomposes time series with periodicity by applying a Loess sequence smoother. As described in Fig. 2, this filter yields three distinct components: a seasonal component capturing periodic evolution, a trend



(a) Semi-major axis (b) Inclination x

Fig. 2: STL decomposition of two equinoctial parameters, during a semi-major axis (a) and an inclination (b) maneuver. The top charts are the raw data (D). The three following are the components of the STL decomposition: Trend (T), Seasonal (S), and Remainders (R).

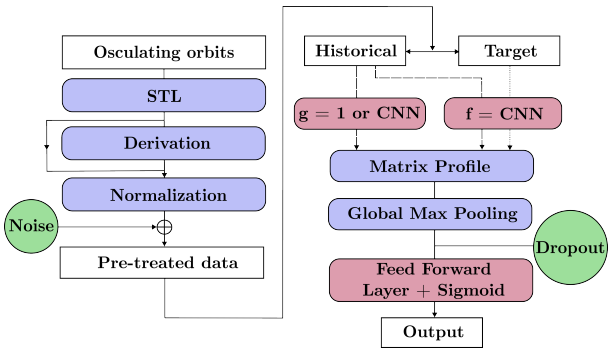


Fig. 3: Pre-treatment and DMP architecture. The blue boxes are non-learnable functions, the red ones denote learnable, and the green circles signify regularization solutions.

component delineating shifts between successive periods, and a remainder component encompassing unexplained information. Subsequently, we derived various features by computing the differences between consecutive timestamps. We standardized the data to ensure uniformity, calculating each satellite's features' mean and standard deviation.

C. Architectures

We propose two architectures (Fig.3) based on a matrix profile computation. They are both composed of a convolutional encoder (the function f) that aims to embed the data in a well-structured latent space for the Euclidean distance computation (Eq.(1)). The matrix profile computation is applied on every latent space dimension and is computed based on the greedy algorithm described in II-B1. The first architecture (DMPh) relies on the original matrix profile definition described in (14), and the second (DMPs) depends on the soft version of the matrix profile described in (15). The function g is either a convolutional network where the parameters are learned during training or equal to one corresponding to the Matrix Profile. Both approaches are set to have a receptive field equal to 120 minutes (the maximum orbital period in LEO). They comprise three layers with a stride equal to three, and each layer doubles the channel size.

IV. Results

A. Implementation Details

Considering that the station-keeping process differs for each satellite, we aim to construct a model that remains resilient to scenario changes. To achieve this, we partition the dataset into train-validation-test sets, ensuring that each scenario is exclusively assigned to one set. The temporal windows $\mathbf{S}^{(i)}$ to classify are 12 hours long; label one if they contain at least one abnormal event and zero if not. The $\mathbf{S}^{(i)}$ are created in a rolling manner along all the scenarios with a step of 72 minutes, which is one-tenth of the window size. The historical data $\mathbf{S}^{\leq i}$ is all the trajectory before it. The dataset

TABLE I: Performances of the two DMP architectures. MA.A stands for Mean Accuracy on sub-class set A. c stands for g as convolutional network and o as g equal to one.

Model	F1-score	Recall	Precision	MA.A	MA.B
DMPh-o	0.35	0.89	0.21	0.84	0.75
DMPh-c	0.19	0.31	0.12	0.72	0.57
DMPs-o	0.73	0.82	0.66	0.9	0.79
DMPs-c	0.75	0.81	0.7	0.91	0.78

exhibits a high imbalance among the classes and within the sub-classes of trajectory types. We delineate two sets of sub-classes. The first (A) distinguishes three different sub-classes: the trajectory without maneuver called *natural*, with station-keeping maneuvers, and with random maneuvers. The second (B) distinguishes six different sub-classes: natural trajectories, station-keeping maneuvers on the semi-major axis, on the inclination, random maneuvers on the semimajor axis, on the inclination, and other random maneuvers. We choose an undersampling strategy for the training according to the subset A with respectively 25%, 25%, and 50% of the dataset. It leads to a balanced dataset according to the label (0:50%, 1:50%). We select the optimal model based on the Area Under the Precision-Recall Curve over 50 epochs, utilizing a batch size of 32 and a learning rate of 0.01, with an adaptive reduction on plateau.

B. Detection performance

We assess the performance of our models on the entire test dataset using the following metrics: Precision, Recall, and F1-score. The performances are also evaluated in function of the sub-classes sets (A and B) via a mean of accuracies on the different subclasses. The precision of the DMPh models is lower than the DMPs one, specifically on the over-represented natural sub-class (cf Fig. 4, which showcases the better-performing results of the proposed DMPs architecture), with a nearly perfect score. Furthermore, the term g, as a convolutional network, seems to improve the DMPs results slightly but penalizes the DMPh architecture. The drop in performance of the mean accuracy on set B compared with set A is due

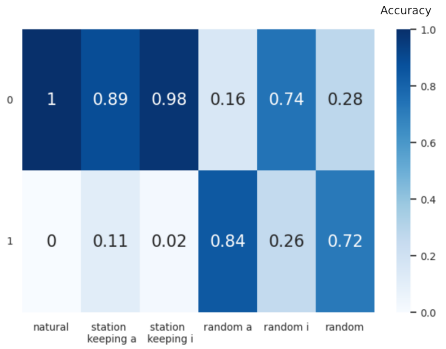


Fig. 4: Confusion matrix of DMPs-c with sub-classes in the x-axis (a stands for semi-major axis and i for inclination) and predictions in the y-axis. The leftmost three columns are labeled 0, and the rightmost three columns are labeled 1.

TABLE II: Performances of the DMPs-c architecture with a matrix profile estimated via STAMP during the inference, according to different fraction values.

Frac	F1-score	Recall	Precision	MA.A	MA.B
1	0.75	0.81	0.7	0.91	0.78
0.1	0.75	0.79	0.72	0.91	0.78
0.01	0.76	0.73	0.79	0.89	0.77
0.001	0.49	0.39	0.68	0.79	0.66

to the low representation of inclination maneuvers, especially for the random ones, and the weak performances of the model to classify random inclination maneuvers as abnormal behavior, as one can see on the Fig. 4. We explain this by under-representing random inclination maneuvers during the training phase. Further, one can notice in Fig. 2 that the inclination maneuvers mainly impact the residues of the STL decomposition, while semi-major axis maneuvers impact both remainders and trend. Because drift in trend contains more information than discontinuities in residues, it could explain the performance difference between semi-major axis and inclination maneuver classifications. The architectures are promising for classifying maneuvers on the semi-major axis as station-keeping ones or not. This characterization is interesting because these maneuvers are the most frequent ones in LEO.

C. Impact of the STAMP algorithm on inference

In the case of real-time detection, using the STAMP algorithm can save considerable amounts of computation. We assess the evolution of performance according to the fraction of queries considered in the STAMP algorithm. In Table II, one notices that the mean accuracies drop when the fraction value is lower than 0.1, which follows the empirical value in the literature. Moreover, reducing the fraction value increases the recall but reduces the precision. This seems intuitive because the model compares the target window with less historical data, which leads to finding less similar behavior in the past. Despite the precision-recall trade-off, the proposed architecture is robust to the STAMP algorithm.

V. Conclusion

This article introduces a novel supervised classification method for analyzing satellite trajectory time series. Our approach extends upon the method proposed by [6], incorporating deep learning techniques. Additionally, we intuit its connection to non-local neural networks. Initial findings show promise in the classification of out-of-station-keeping maneuvers. In future studies, we intend to investigate alternative decomposition methods to enhance the characterization of inclination maneuvers.

Acknowledgment

This work was supported by the French Defense Innovation Agency (Cifre-Défense 2021/0003/AID). This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014111R1 made by GENCI.

References

[1] D.A. Vallado, *Fundamentals of Astrodynamics and Applications (Vol. 12)*, Space Technology Library, springer science & business media edition, 2007.

[2] Charlotte Shabarekh, "Efficient object maneuver characterization for space situational awareness," *32nd Space Symposium, Technical Track*, p. 7, 2016.

[3] Thomas G Roberts et al., "Geosynchronous satellite maneuver classification via supervised machine learning," *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2021.

[4] Varun Chandola, "Anomaly Detection : A Survey," *ACM computing surveys (CSUR)*, p. 74, 2009.

[5] Ane Blázquez-García et al., "A Review on Outlier/Anomaly Detection in Time Series Data," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–33, Apr. 2022.

[6] Chin-Chia Michael Yeh et al., "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 1317–1322, IEEE.

[7] Qian Liu et al., "A Novel Matrix Profile-Guided Attention LSTM Model for Forecasting COVID-19 Cases in USA," *Frontiers in Public Health*, vol. 9, 2021.

[8] Hieu X. Nguyen et al., "MPCNN: A Novel Matrix Profile Approach for CNN-based Sleep Apnea Classification," Nov. 2023.

[9] Chin-Chia Michael Yeh et al., "Ego-Network Transformer for Subsequence Classification in Time Series Data," Nov. 2023.

[10] Ashish Vaswani et al., "Attention is All you Need," *Advances in neural information processing systems*, 2017.

[11] Qingsong Wen et al., "Transformers in Time Series: A Survey," May 2023.

[12] Haixu Wu et al., "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

[13] Abdullah Mueen et al., "Fast approximate correlation for massive time-series data," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, Indianapolis Indiana USA, June 2010, pp. 171–182, ACM.

[14] Xiaolong Wang et al., "Non-local Neural Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018, pp. 7794–7803, IEEE.

[15] Antoni Buades et al., "Non-Local Means Denoising," *Image Processing On Line*, vol. 1, pp. 208–212, Sept. 2011.

[16] Sheng Zhong et al., "MASS: Distance profile of a query over a time series," *Data Mining and Knowledge Discovery*, Feb. 2024.

[17] Stefan Baudier et al., "Synthetic Dataset of Maneuvering Low Earth Orbit Satellite Trajectories for AI Analysis," 2024.

[18] R. A. Broucke et al., "On the equinoctial orbit elements," *Celestial Mechanics*, vol. 5, no. 3, pp. 303–310, May 1972.

[19] R.B. Cleveland et al., "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," *International journal of biometeorology*, vol. 6, no. 1, pp. 3, 1990.