On the integration of large scale time series distance matrices into deep visual analytic tools

Inmaculada Santamaria-Valenzuela[1*],
Victor Rodriguez-Fernandez[1] and David Camacho[1]

[1]Computer Systems Department, Universidad Politécnica de Madrid, Madrid, 28031, Spain.

*Corresponding author(s). E-mail(s): mi.santamaria@upm.es;
Contributing authors: victor.rfernandez@upm.es; david.camacho@upm.es;

Abstract

Time series are essential for modelling a lot of activities such as software behavior, heart beats per time, business processes. The analysis of the series data can prevent errors, boost profits, and improve the understanding of behaviors. Among the many techniques available, we can find Deep Learning techniques and Data Mining techniques. In Data Mining, distance matrices between subsequences (similarity matrices, recurrence plots) have already shown their potential on fast large-scale time series behavior analysis. In the Deep Learning, there exists different tools for analyzing the models embedding space for getting insights of the data behavior. DeepVATS is a tool for large time series analysis that allows the visual interaction within the embedding space (latent space) of Deep Learning models and the original data. The training and analysis of the model may result on a large use of computational resources, resulting in a lack of interactivity. To solve this issue, we integrate distance matrices plots within the tool. The incorporation of these plots with the associated downsampling techniques makes DeepVATS a more efficient and user-friendly tool for a first quick analysis of the data, achieving runtimes reductions of up to $\mathbf{10^4}$ seconds, allowing fast preliminary analysis of datasets of up to 7M elements. Also, this incorporation allows us to detect trends, extending its capabilities. The new functionality is tested in three use cases: the M-Toy synthetic dataset for anomaly detection, the S3 synthetic dataset for trend detection and the real-world dataset Pulsus Paradoxus for anomaly checking.

Keywords: Time Series Analysis, MPlot, Visual Analytics, Machine Learning, Deep Learning

1 Introduction

The modelling and analysis of the obtained data is essential in so many areas for the understanding, boosting and prevention of different behavior of events. For instance, the detection and prevention of rare events such as anomalies in oil-producing wells can prevent detrimental financial implications [1]. The modelling can also be useful in the resources optimization in nature disasters [2], and behavior analysis [3–7]. Time series are the main tool used for the modelling and analysis of the data [8–12] as they allow an easy way to relate events to specific times and are useful to predict future events which can be helpful, for example, for predicting orbits in space objects [13]. The analysis of time series can be done for many different perspectives. For instance, for anomaly detection we can use techniques from Classic Machine Learning, Signal Analysis, Sthocastic Learning, Statistcs, Deep Learning, Data Mining or specific Outlier Detection techniques [14]. This work focuses on combining the strength of Data Mining and Deep Learning techniques for a better interactive analysis of large scale time series.

In the field of Deep Learning (DL), significant effort has been made in understanding the embedding space produced by the different algorithms. There are a lot of tools for interactive visualization of embedding spaces for datasets analysis [15, 16]. Among these tools, DeepVATS (Deep Visual Analytic tool for Time Series [17]), stands out as an excellent tool for analyzing the embedding space of large scale Time Series, combining DL algorithms with visual tools. DeepVATS is based on training a DL model for the later analysis of its embeddings spaces, as its structure represents in a vectorized way the structure of the time series, giving a more intuitive comprehension of it. This process involves a large time consumption [18], so the interactivity is reduced as the size of the time series grows. To avoid this waiting time, we can take part on the emerging research on foundation models that joins Transformer and Large Language Model techniques from with time series analysis greatly reducing the execution time while getting the embedding space for specific datasets [19, 20].

In the field of Data Mining (DM), distance matrices and its associated methods such as SCAMP [21] or SCRIMP++ [22] and tools shuch as MPlot (similarity matrix plot, showing the distance between subsequences of the time series) [23] stands out as a great approach for time series analysis [14]. Although, as will be shown in future sections, the computation of this plots require a lot of computational resources resulting even on Out Of Memory errors, they have been scaled up to perform large scale time series data analysis [24]. Thus, they provide an efficient way for fast and resource-efficient analysis of the different time series behaviors such as anomalies, patterns or novelets. The use of MPlot can give a good overview of time series patterns even when time series are downsampled for reduced resources neccesity [25]. Thus, MPlot help in a first view of the time series while waiting for the Deep Learning training.

The present research focuses on the integration of MPlot as an useful techique from Data Mining into DeepVATS as an interactive visual application for Deep Learning Analysis. This combination results on an enhancement in univariate large scale time series behavioral analisis, allowing a quick preview of the results and the possibility of detecting trends withing the visual app. The detection of trends is the result of the representation of the distance between the subsequences of a concret size across the time series. So, if the time series has a trend, the distance will get bigger as the initial

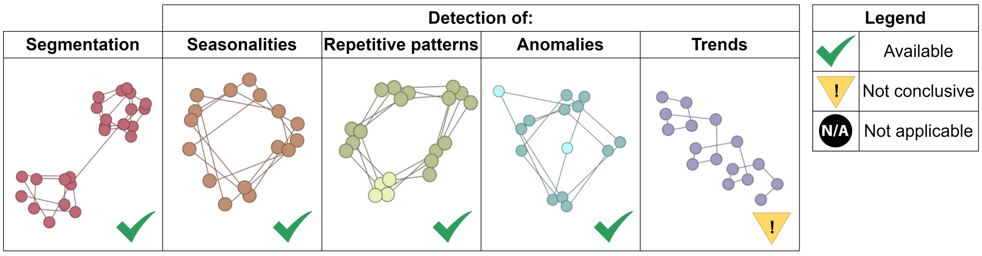

Fig. 1: Summary of DeepVATS capabilities for time series tasks, adapted from [17].

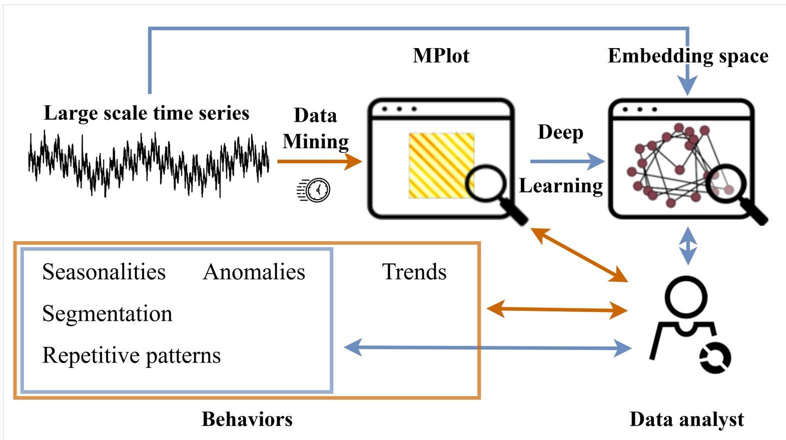

Fig. 2: Flow of DeepVATS with the integration of the similarity matrix plot (MPlot) for preliminary analysis of large time series. The image have been adapted from [17]. In blue, the original flow: the time series is loaded and analyzed through Deep Learning to obtain the embedding space, which is then explored by the analyst to detect diferent time series behaviors. In orange, the prposed flow when MPlot is integrated, allowing a fast preview of the analysis through data minin techniques to quickly identify time series behavior (adding the detection of trends).

timesteps of the subsequences get more separated. Thus, this integration overcomes the initial lack of DeepVATS on distance analysis (see Figs. 1, 2). To analyze the contribution, three different datasets have been selected as cases of use:

- The M-Toy synthetic dataset from STUMPY [26] allow to evaluate the detection of anomalies within Data Mining technique, resulting on a fast-check of the results obtained when comparing to the previous evaluations [17].
- The S3 dataset is a synthetic dataset [17] that reproduces an increasing trend time series, allowing the analysis of MPlots in trend detection.
- The Pulsus Paradoxus dataset is a real-world dataset previously analyzed [23] that allow us to compare the fast use of the different techniques on detecting patterns.

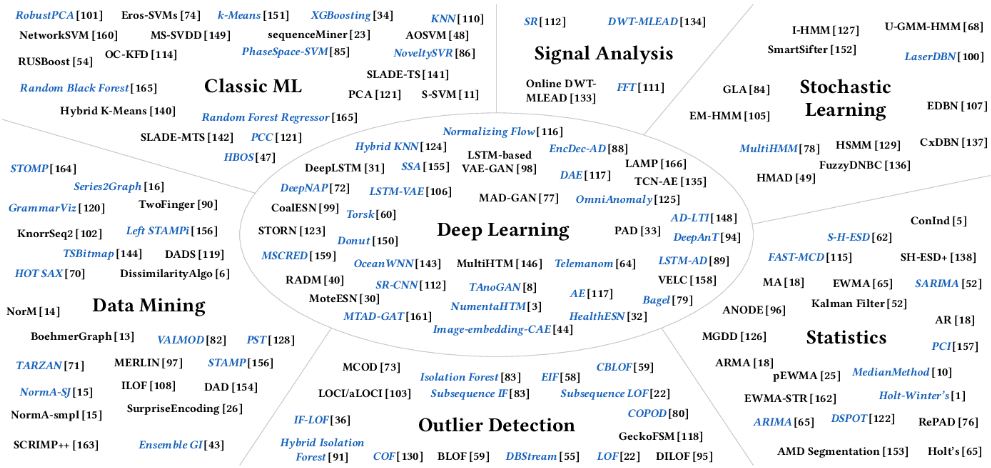

Fig. 3: 158 anomaly detection methods for time series data structured by their methods family collected by Sebastian et al. [14].

This dataset will be used for the memory and time consumption analysis as it has more than 7M elements.

The next sections are organized as follows: section 2. Background; introduces the theoretical concepts for MPlot comprehension. Section 3. Integration of MPlot into DeepVATS; introduction to DeepVATS and integration with Mplot. Section 4. Conclusions and future work; summary of research and expectations.

2 Background

The exploration of time series data is essential in many areas. Obtaining insights from time series can become challenging as the volume of data grows.

DeepVATS is a tool that assists in the main tasks of time series analysis in a visual and interactive way (see Fig. 1). There are many different ways to analyze time series. Fig. 3 shows a classification that can be extended to all tasks. The families can be summarized into four: Statistical techniques (e.g. AutoRegressive Integrated Moving Average models - ARIMA [27]. exponential smoothing models [28]), Machine Learning (e.g. MOMENT family [29]), Hybrid methods that integrate both statistical techniques and machine learning methods to detect anomalies in time series [30]; and Data Mining, based on obtaining valuable information by collecting and comparing data from large datasets.

DeepVATS is currently based on a Masked Time Series Autoencoder (MTSAE). This Deep Learning algorithm learns compact representations (embeddings) of time series by masking and reconstructing parts of the data, which allows for the identification of patterns and anomalies. The goal in this paper is to add the MPlot, a graphical view of the distances between time-series subsequences that is the basis of some of the methods in the Data Mining family (STOMP [31], STAMP [32], MERLIN [33], MADRID [34], SCRIMP [22]). These algorithms perform well in the analysis of univariate time series (see Tb. 3 in [14]). These algorithms use, in particular, the matrix profile. which is a feature built from the MPlot that has been referred to as a Swiss army knife for its capabilities in the detection of time series behavior [35].

This section introduces the MPlot tool and the MatrixProfile feature, describes the DeepVATS tool, and shows how to integrate MPlot into DeepVATS to get the best from both.

2.1 MPlot: similarity matrix plot

The work presented in this study aims to add an interactive exploration of MPlot integrated into DeepVATS. In this way, you can analyze the plot while waiting for the MVP model to be trained and get two different perspectives for the analysis. MPlot is a plot based on the distances between the subsequences of a time series. Let us introduce the different definitions necessary for the comprehension and implementation of the plot, following [23].

Definition 2.1: Time series and subsequences

A time series $T = \{t_i\}_{i=0}^{n} = \{t_0, t_1, \ldots, t_n\}$ is a sequence of real-valued numbers, $t \in \mathbb{R}$. We call subsequence to an ordered subset of T: $T^{(k,m)} = \{t_i\}_{i=k}^{m-k+1}$, following the same order than the original time series.

See that in Def. 2.1 the subsequence starts at the position k with a length of m time steps. This value (m) is called subsequence length and, in the same way, n is called time series length. This will be notated as $T \sim n$, $T^{(k,m)} \sim m$. The subsequence length will also be called "window length" following the analog definition for MTSAE.

reacNow, we can define the distance between two subsequences of T of the same length by using them as vectors: $d(T^{(k,m)}, \hat{T}^{(l,m)}) = d((t_i)_{i=k}^{m-k+1}, (\hat{t}_i)_{i=l}^{m-l+1})$. The distance most widely used is the *z-normalized Euclidean distance* [32].

Now, we can define the distance profile and the distance matrix or similarity matrix based on the distances of the subsequences of size "window length".

Definition 2.2: Distance Profile

Given $T_A \sim n_A$, T_B, the distance profile is the vector of distances between each subsequence in T_A and $T_B^{(k,m)}$:

$$DP_{A,B}^{(k,m)} = (d(T_A^{(i,m)}, T_B^{(k,m)}))_{i=0}^{n_A-m+1}.$$

Definition 2.3: Similarity Matrix

Given $T_A \sim n_A$, $T_B \sim n_B$. The *similarity matrix* or *distance matrix* is the matrix $DM_{A,B} \in \mathbb{R}^{n\_rows \times n\_cols}$ where each row is a distance profile with the corresponding T_B subsequence. That is, $DM_{A_B} = (DP_{A,B}^{(k,m)})_{k=0}^{n_B-m+1} = ((d(T_A^{(i,m)}, T_B^{(k,m)}))_{i=0}^{n_A-m+1})_{k=0}^{n_B-m+1}$, with $n\_rows = n_A - m + 1$ and $n\_cols = n_B - m + 1$.

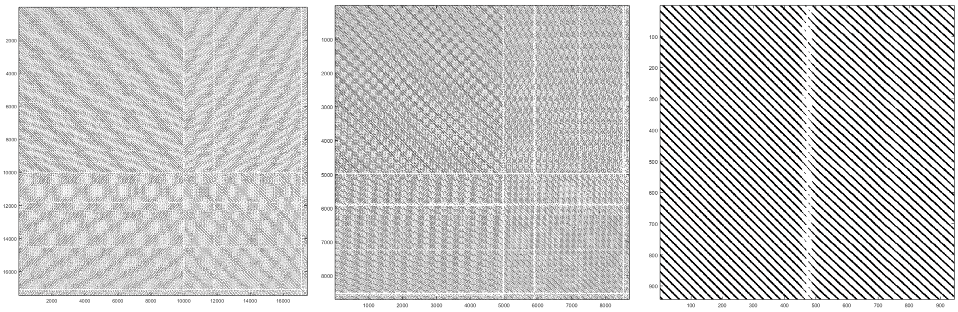

Fig. 4: MPlot for the Pulsus Paradoxus SPO2 time series, using sequence length 54 (1 pulsus length). On the left, the complete time series is used. On the middle, the time series is subsampled getting the odd elements ($ts[::2]$). On the right, a zoom in the 10k timestamp.

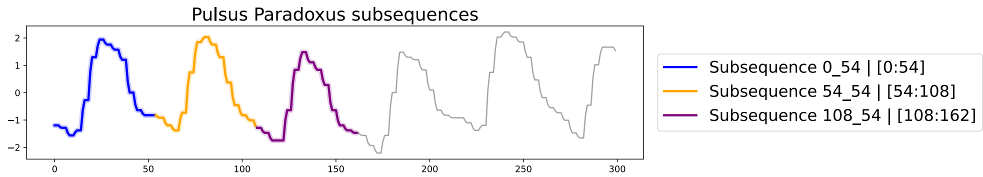

Fig. 5: First Pulsus Paradoxus data subsequences of size 54.

The Similarity Matrix plot or MPlot is the plot resulting of visualizing in a heatmap mode that matrix (see fig. 4). This matrix computation can incur a very large memory and time cost. Three ways can be followed to make it more efficient. One possibility is to compress the time series using techniques such as Piecewise Aggregate Approximation [25] to reduce the ammount of points to analyze. Another option is to use the Matrix Profile of minimums of distances (see Def. 2.4). This vector computation has been widely optimized with algorithms such as STAMP [32] or SCRIMP++ [22]. Finally, different algorithms can be applied for partially computing the MPlot or zooming-in zooming-out. You can check some of these techniques (for example, SPLAT) in Shahcheraghi et al. [23].

Definition 2.4: AB-join Matrix Profile

An *AB-join Matrix Profile* $MP^m_{A,B}$ is a vector of euclidean distances between each subsequence of T_A and its nearest subsequence of T_B; that is, $MP^m_{A,B} = (min(DP^{k,m}_{A,B}))_{k=0}^{n_B-m+1}$. If $A = B$, the matrix is called *self-join Matrix Profile*.

2.2 MPlot example: Pulsus Paradoxus

Shahcheraghi et al. [23] show different examples of how an MPlot can give initial insights on time series data previous to a deeper analysis. It aids to find for different types of behaviors such as patterns, discords or novelets (emerging patterns you didn't see before), shapelets (for classification tasks).

One example is the Pulsus Paradoxus. Pulsus paradoxus is an abnormal decrease in systolic blood pressure of more than 10 mmHg during inspiration. This condition can indicate serious medical problems such as tamponade [36]. The time series "PulsusParadoxusSP02_30_10000"[1] contains the SPO_2 (oxygen saturation) along 17521 time steps, being 54 the length associated with 1 heartbeat (1 pulsus; see Fig. 5).

By checking the complete matrix plot (Fig. 4) we can see that when the time series T_A and T_B have the same length, the MPlot is symmetric. This is true as distance functions are always symmetric ($d(u, v) = d(v, u)$). This property can be used for better memory and time compsumtion.

As mentioned, each heartbeat takes approximately 58 time steps, which means that $20,000$ time steps are approximately 344 heartbeats, which is the number of diagonals you can see in the MPlot. The reason is that the SPO_2 metric repeats the same pattern in each pulsus, when we take the distances, the sequences starting in $58 \cdot n$ with $n \geq 0$ will be very close between them, and so on. Thus, each diagonal corresponds to one beat pattern. There are also white spaces. The white breaks show that in that place the subsequences are different, that is, the pattern breaks. According to the MPlot pattern catalog [37] this behavior means that "The same pattern is reoccurred with an unseen pattern in between".

If we use the Matrix Profile to look for a pattern, we can find a motif (Fig. 8). If we take it, we can see that it is really similar to the one in the middle of the red sequence (Fig. 9). This may be the "unseen pattern". But, as they mention in [23], it seems that something happens approximately every 8 beats (see Fig. 7). Let us see what happens in the blue subsequence (Fig. 10). Interestingly, in the orange positions, we observe that for every eight/nine pulses, there is a heartbeat where the value does not quite reach the "minimum" but remains slightly higher. Meanwhile, the rest of the time series closely resembles the red sequence. Something similar should happen if we look to the rest of the time series. According to Shahcheraghi et al. [23], the 8 pulsus ratio is due to the relation 1 respiration cycle per each 8 heartbeats. The anomaly detected in $10K$ is due to heart damage (known as tamponade) made during a surgery that reduces the efficiency of the person in producing oxygenated blood (see Fig. 7).

2.3 DeepVATS

DeepVATS [17] is a Visual Analytics (VA) tool that uses Deep Learning to extract valuable information from time series (TS) through interactive embedding projections and time series plots. It uses dimensionality reduction techniques for efficient analysis and interaction with large time series.

It is based on three main modules (see Fig. 11): the Deep Learning Module, the Storage module, and the Visual Analytics module. Sometimes, I will refer to the first two module as "backend" and to the visual app as "frontend". The next subsections briefly describe each module.

[1] Data set available at https://drive.google.com/file/d/1mJs_FSjSnffw2xPJhu3SlES2kXSCMoNy/view

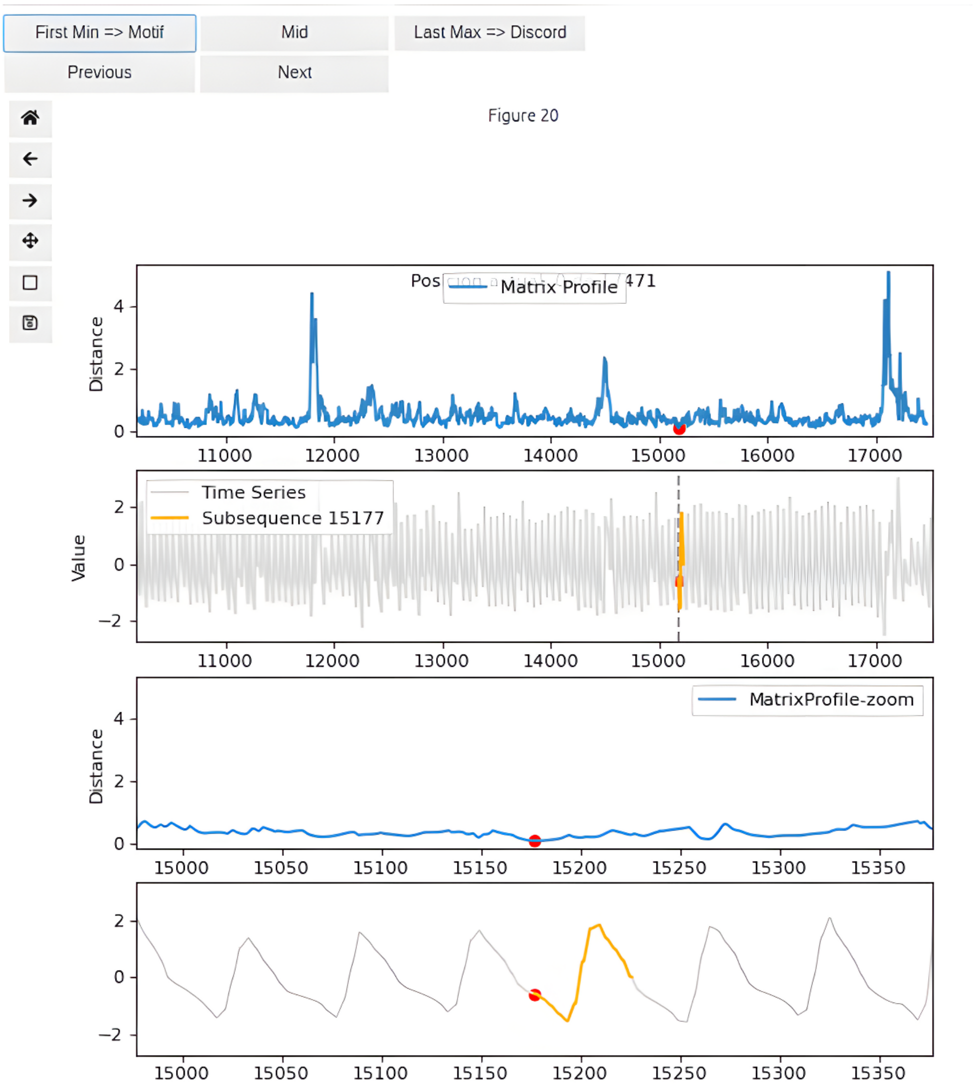

Fig. 6: Pulsus Paradoxus example Matrix Profile interactive plot. The plot shows the minimum value in the Matrix profile (i.e., the motif).

2.3.1 Deep Learning Module

This module is responsible for all the data processing and computations. Here you can:

- Load your data set from both local files and the Weight and Biases database [38]. It allows basic transformations for normalizing the data or selecting different columns in case you have multivariate time series.

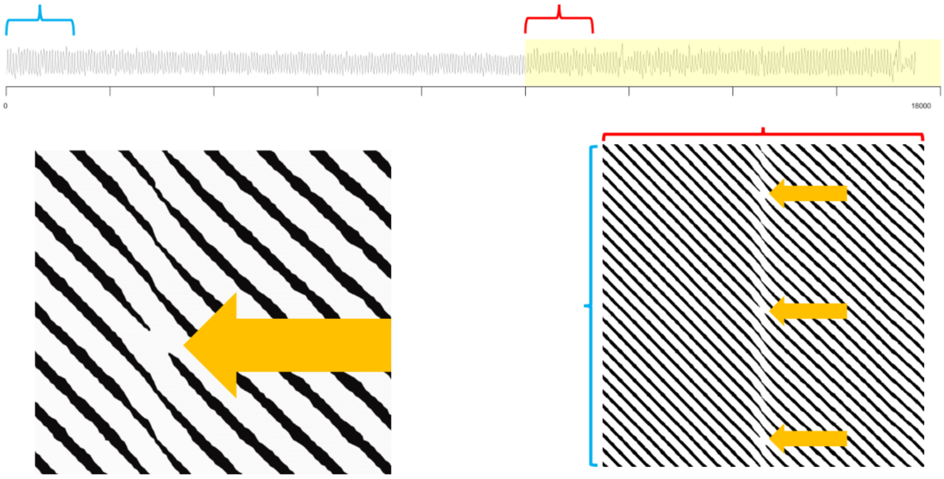

Fig. 7: Pulsus Paradoxus SPO2 MPlot using SinMat Shahcheraghi et al. [23]. The image is obtained from the notes on https://drive.google.com/file/d/1fjWUzVQf-8XmS5epDa_ulm4-bDX51Vxv/view. It shows a new pattern ocurred each eight pulsus that breaks the most repeated one.

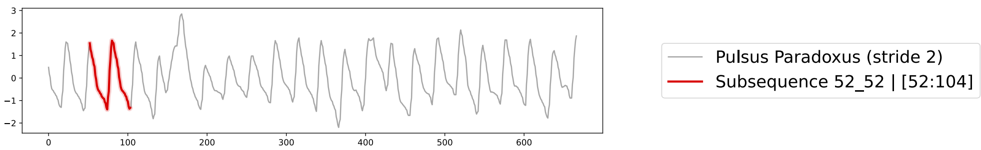

Fig. 8: Pulsus Paradoxus motif found while looking for the motif using the Matrix Profile. The plot starts at $t_0 = 11530$, with the motif subsequence starting at $t = t_0 + 52 * 2$ with a length of $52 * 2$ (taking into account that the time series it loaded in steps of 2).

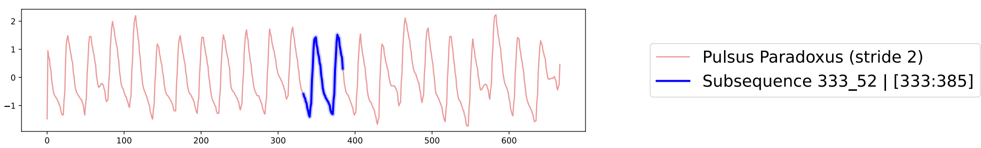

Fig. 9: Pattern found in [23]: red subsequence starting from $t = 10666$ with length 52. The plot starts in index $t = 10000$ and uses an stride of 2 between timestamps.

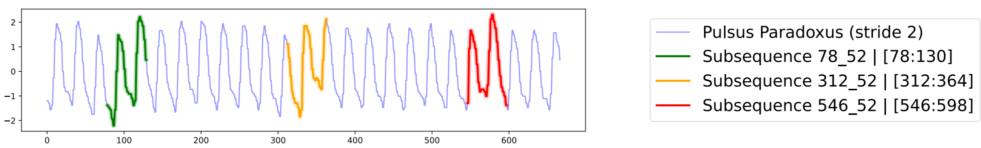

Fig. 10: Pattern found in [23]: blue subsequences, starting from $t \in \{208, 624, 1092\}$ with length 52. The plot starts in index $t = 0$ and uses an stride of 2 between timestamps.

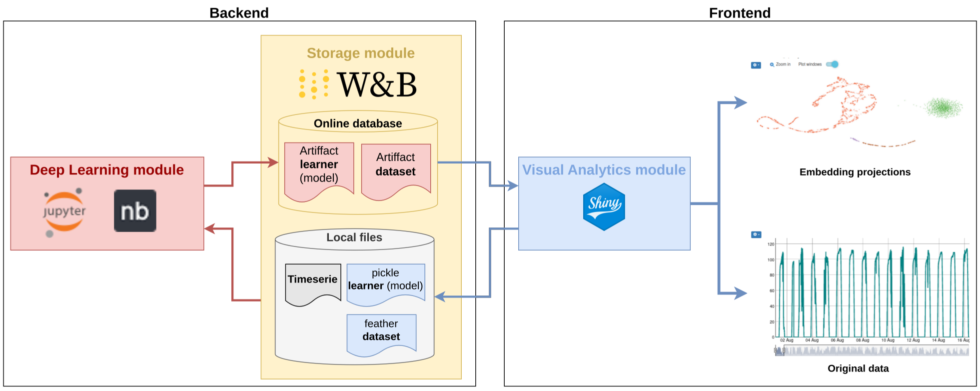

Fig. 11: Deepvats work-flow diagram

- Train the deep Learning module for obtaining the embeddings. This step is based on a Masked time series AutoEncoder (MTSAE) based on the timeseries AI's MVP method [39]. The trained model with the resulting embeddings is saved by the Storage Module.
- Reduce the embedding dimensions. You can use:
  - Uniform manifold approximation and projection (UMAP) [40], which is frequently used for analyzing the embedings [15].
  - t-distributed Stochastic Neighbor Embedding (t-SNE) [41], used, for example, in the TensorFlow's embedding projector [16].
  - Principal Component Analysis (PCA). It is being integrated as a first step before computing UMAP to ensure the integrity of the obtained clusters.

2.3.2 Storage Module

As shown, the storage module is a simple way to store the data and communicate it between the back-end and the front-end ensuring a good flow of the data within the computation. It is based on your local file system and the Weight and Biases database [38], which can also be used to track experiments, as it keeps logs of executions, including specific metrics such as the percentage of GPU use.

2.3.3 Visual Analytics Module

This is the visual part of the tool and the one that you are meant to use for interactive analysis of the app. In this window, you can interact with the embedding projections plot and the time series data plot and select different parameters for the computations (see Fig. 12).

The installation instructions are shown in the README file in the repository (https://github.com/vrodriguezf/deepvats/blob/master/README.md). The execution will be shown within the examples in Section 3. Integration of MPlot into DeepVATS.

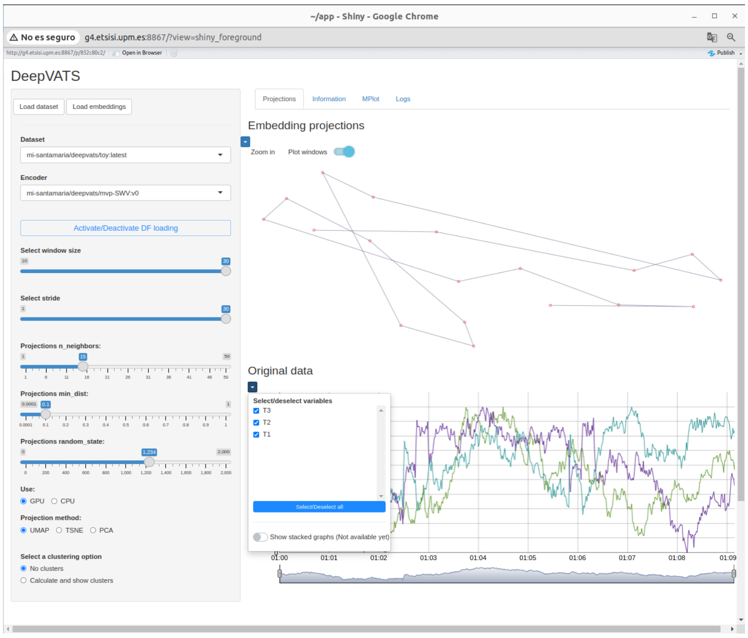

Fig. 12: Example of execution in the Visual Analytics app.

3 Integration of MPlot into DeepVATS

This section focuses on how to use DeepVATS to generate and visualize an MPlot and its relation with the information found within the embedding projection plot. The main goal is to add the MPlot to the visual app and make it interactive for a good analysis. A previous efficiency evaluation is performed.

3.1 Efficiency evaluation of MPlot-DeepVATS

The computation of MPlot requires substantial computational resources and execution time, especially as the length of the time series increases. As shown in Table 1, we run into memory error when analyzing time series from 493K elements. Memory consumption can be addressed in different ways like decreasing the number of decimals, modifying window length or reducing time series length. Following previous solutions and trying to not to affect the analysis, the time series can be compressed before the computation. The MPlot of the time series downsampled to 10k points, combined with specific zoom techniques, still provide valuable insights into the global time series Shahcheraghi et al. [23]. Therefore, we compute the MPlot using a maximum number of points (by default, 10k), resulting in significantly reduced processing times of 5 to 7 seconds for an initial view of the time series behavior. In addition, Matrix Profile-based algorithms have demonstraated good performance in different domains [42–44]. For these reasons, the Matrix Profile and the MPlot have been selected to address the shortcomings in DeepVATS. We have included interaction between the MPlot, the matrix profile plot, and the time series plot and checked the relation to the results

Table 1: Computation time for DeepVATS first analysis, both before and after incorporating MPlot.

Time Series		Execution Time (seconds)		
Frequency (seconds)	Number of elements	DeepVATS	DeepVATS + Mplot (Full)	DeepVATS + MPlot (10K)
4	7,397,222	144,850.00	OOM Memory error	6.93
20	1,479,445	29,075.00	OOM Memory error	7.08
60	493,149	9,600.00	OOM Memory error	4.98
300 (5m)	98,630	1,885.71	546,06	7.42
600 (10m)	49,315	900.00	46,34	7.30

obtained within the MVP method. Table 1 shows the computation time necessary to analyze the time series behavior before and after adding MPlot to DeepVATS. This table is based on the execution of DeepVATS for the Solar Power Dataset (4 Seconds Observations) dataset [45]. This data set has been downsampled following the guidelines proposed in [18]. This downsample results in five time series with frequencies from 4 seconds to 10 minutes (first column) and 50K to 7.4M elements (second column). The third column of the table shows the execution time necessary for training the MVP model with an stride of 1, 100 epochs, and window lengths from 460 to 100. The fourth column shows the execution time needed for computing both the Matrix Profile (using STUMP) and the MPlot (using SCAMP) for the full time series. The fifth column shows the same computation but downsampling the time series to approximately $10K$ elements using Piecewise Aggregate Approximation. These executions show that the full MPlot cannot be used directly by the tool due to the intensive memory usage required for calculations and storage, although the execution time for small time series may be faster than MVP training. However, the execution of the downsampled time series MPlot is really fast, thus being a great tool for previewing the time series for a further analysis.

Thus, the integration of MPlot into DeepVATS expands its usability as it allows another perspective on patterns and anomaly detection. In addition, this integration provides an interactive way to analyze MPlot within the Matrix Profile and the time series. The only thing similar to this integration that we have found is the MPlot Explorer[2], by Zach Zymmerman, which has been the baseline for our code alongside the instructions in [23] and the basis code for computing and visualizing MPlot[3], which has the problem that it is implemented in MATLAB. There is a version in the library SCAMP based on it's same name MP algorithm [21], however, it warns for experimental code in some parts. So, we will use this library and the widely extended STUMPY [26] as well as naive brute-force algorithms to provide a flexible library while new implementations in Python emerge.

[2]Available in https://github.com/zpzim/mplot-explorer.

[3]Available in https://www.cs.ucr.edu/~eamonn/MatrixProfile.html and https://sites.google.com/ucr.edu/mplots/codes?authuser=0.

It is especially important to note that the plots are done following the MATLAB convention to ensure the integrity with the previous work. Thus, you must be careful when analyzing the ordinate (y) axis. The origin will always be at the top of the image instead of at the bottom (see Figure 4).

Taking those details into account, the Distance Profile, Distance Matrix, Matrix Profile, and Matrix Profile Plot (MPlot) computation and plotting have been implemented to get a basic code for computing and visualizing them.

3.2 Using MPlot-DeepVATS in specific Use Cases

This subsection focuses on how to use the new MPlot-DeepVATS tool, and how it can be used to generate, analyze, and visualize new patterns in some specific use cases (M-Toy, S3, and Pulsus Paradoxus). As shown in Fig. 7, the interface of MPlot-DeepVATS allows you to easily view the MPlot, the Matrix Profile, and the compared time series all within a single tab. You can select the variable for analysis (if your time series is multivariate), specify the maximum number of points to compute for the MPlot, and adjust the subsequence length (window size). Instead of directly selecting columns and rows, you interact with the time series; zooming in with the selector automatically updates the MPlot. The Matrix Profile is also interactive, enabling you to explore motifs and discords and view their positions in the MPlot, as well as the corresponding subsequences in the time series below. Additionally, you can select other parameters, including computation methods, making the workflow seamless and fully accessible within the visual app. Below, some selected use cases are shown, to demonstrate how the new tool can be used to identify and discover patterns in the time series analyzed.

3.2.1 Use Case 1 - M-toy

Let us compare the MPlot with a known embedding projection plot: $M - Toy$. This is a synthetic 3-dimensional time series analyzed in [17] gathered from the STUMPY repository [26]. This time series has three variables with two known anomalies that will allow us to test MPlot for the task of anomalies detection. The main problem with M-Toy is that it is multivariate, but MPlot can only work with 1-column data. However, we know that $T3$ is the less representative variable when trying to detect the two main anomalies of the time series (see Fig. 15). Thus, we have generated the MPlot for each variable of the time series. However, it does not give a clear visualization of the anomalies as there is no clear diagonal break. By checking the catalog [4] all we can say is that we are probably seeing a lot of repeating patterns with decreasing frequency. The only thing that is equal in 13 is that there is a large dark zone that reduces to a diagonal line just before reaching the anomaly indexes. This makes the idea that some parts of the time series have been really similar in that zone and suddenly they start to be different. In the case of $T1$ and $T2$ (Fig. 13), the $T1$'s plot has a diagonal break in the 150 timestep (horizontal time series), which is some kind of advice of rare motif or anomaly. However, it does not seem to be a good idea to try to analyze a single MPlot to gain insight into multivariate time series.

[4] UCR catalog for understanding MPlot https://sites.google.com/ucr.edu/mplots/catalog?authuser=0

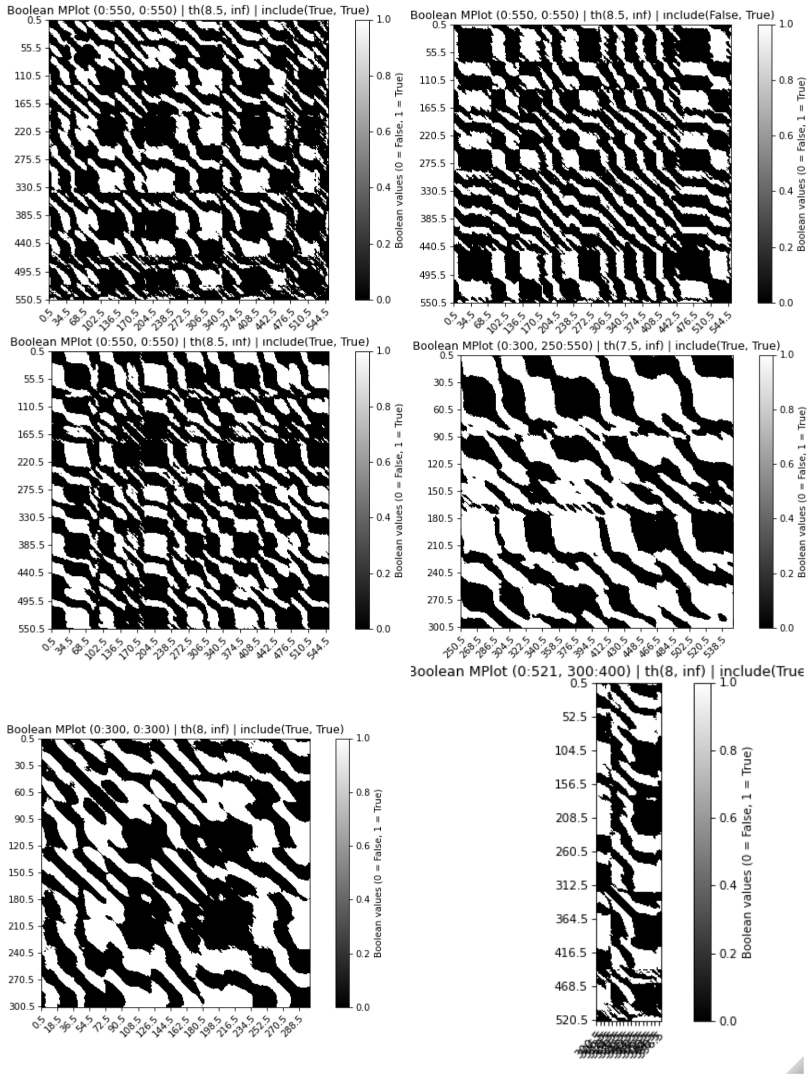

Fig. 13: MPlot for the M-Toy dataset variables. The first row shows the T1 (left) and T2 (right) variables. The second row displays different zoom levels for the T1 variable. Finally, the third row presents different zoom levels for the T3 variable.

3.2.2 Use Case 2 - Looking for trends, S3

In [17] it was shown that DeepVATS was not especially good at detecting trends. To check how the use of MPlots affects the detection of trends, the "S3" time series has been analyzed. This time series is a synthetic univariate time series spanning 20 days of data generated with parameters that includes a time-dependent increasing trend (see Fig. 17).

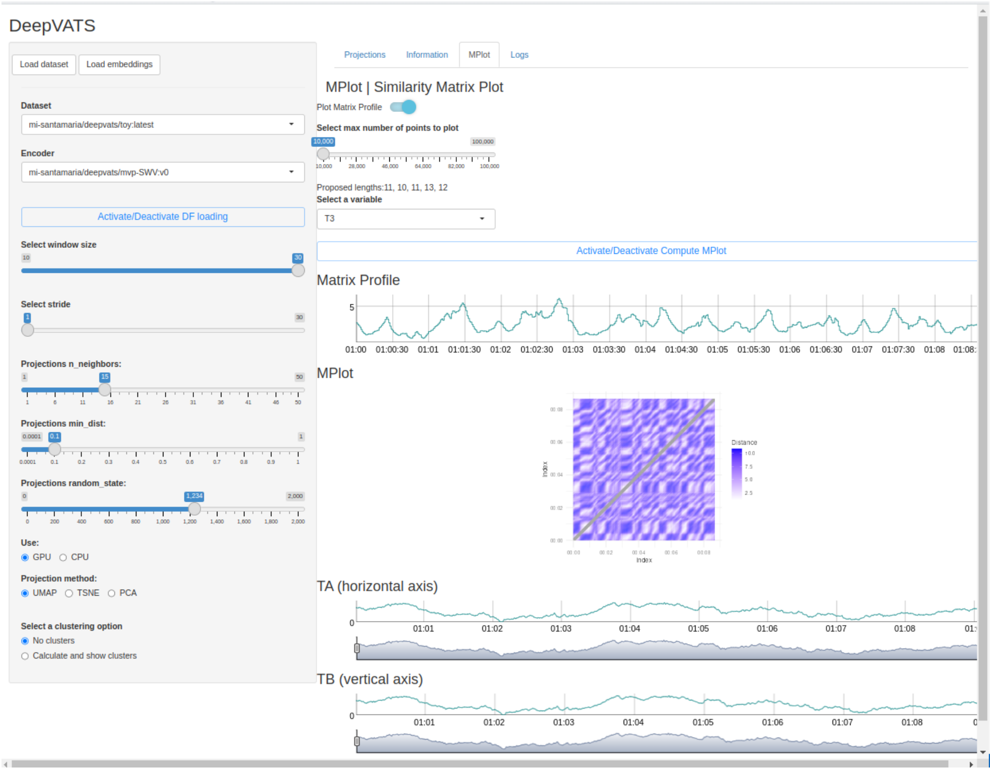

Fig. 14: M-Toy MPlot in the Visual App.

The Euclidean distance is useful for identifying trends in a time series because it preserves the magnitude of the data. In contrast, the z-normalized distance is not suitable for this purpose as it standardizes the data, thus deleting the inherent value information (but the repeated pattern that changes its height is easier to see in the z-normalized version).

SCAMP can only be computed using the z-normalized Euclidean distance since it does not have a parameter for the distance. Fig. 16 shows this effect by running the MPlot time series with the Euclidean distance for STUMP and the z-normalized Euclidean distance for SCAMP.

However, we cannot know with a distance function if this trend is upperward or downward, as they are symmetric: the distance will be the same whether you go up or down. But in DeepVATS we can compute the Distance Matrix using brute-force; this version does not ask for properties to the "distance" function. Thus, we can try to define a specific function to detect trends, so we get a "pseudo-MPlot". The most basic: $d(\vec{u}, \vec{v}) = \mu_u - \mu_v$. As Fig. 17 show, the MPlot generated with this function clearly shows the upward and downward trends. Also, we could still save only one of the two main triangular submatrices of the pseudo-MPlot in a triangular matrix, since $d(u, v) = -d(v, u)$.

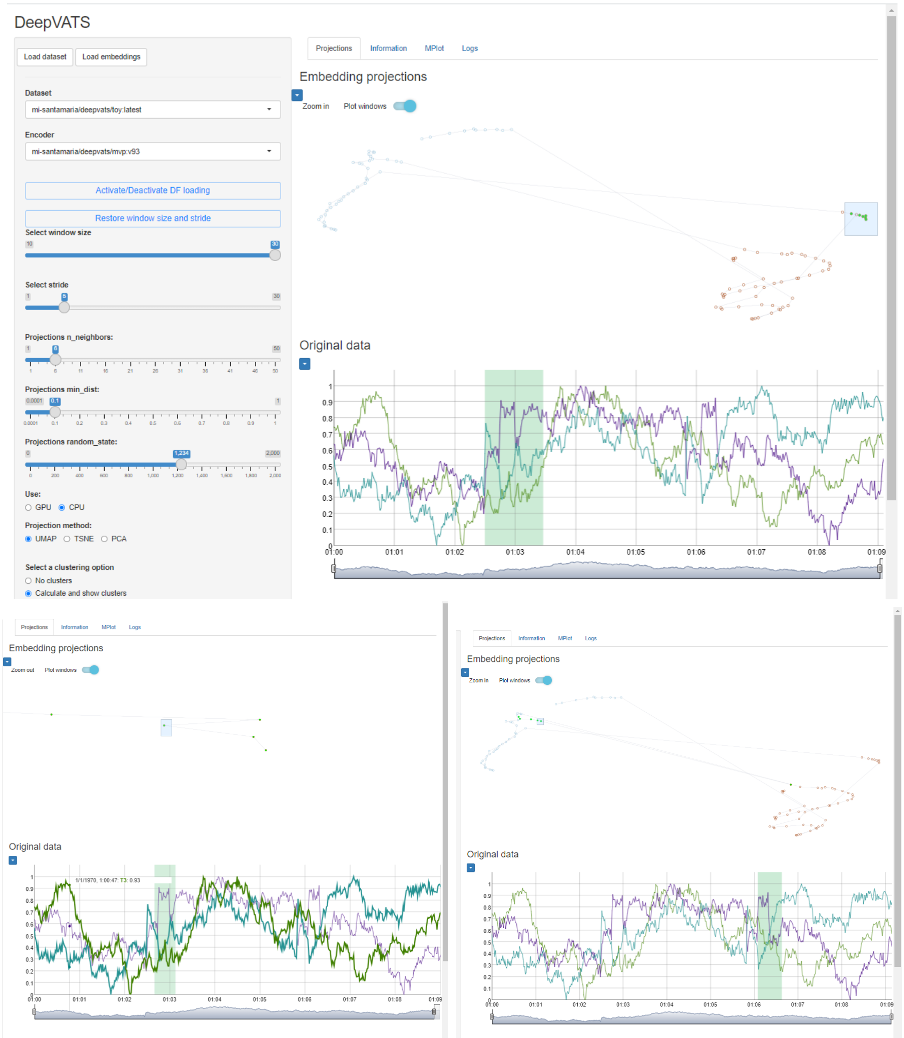

Fig. 15: Embedding space analysis for the two anomalies of the M-Toy time series.

3.2.3 Use Case 3 - Pulsus Paradoxus

The third use case is done using the Pulsus Paradoxus SPO_2 dataset, analyzed in Shahcheraghi et al. [24]. It is important to remark that you must check the DeepVATS execution pipeline to train the model before trying to analyze the Pulsus Paradoxus embedding space in the visual app. The DeepVATS Pipeline is followed to analyze the dataset; that is: initialize and configure Artifact, load the dataset to W&B, and train the data set executing the related notebooks, and open the app and check both the MPlot and the embedding space.

The interactions with the Matrix Profile are shown in 2. Background. After some interactions with the pulsus paradoxus embedding space (see Fig. 18), the conclusion

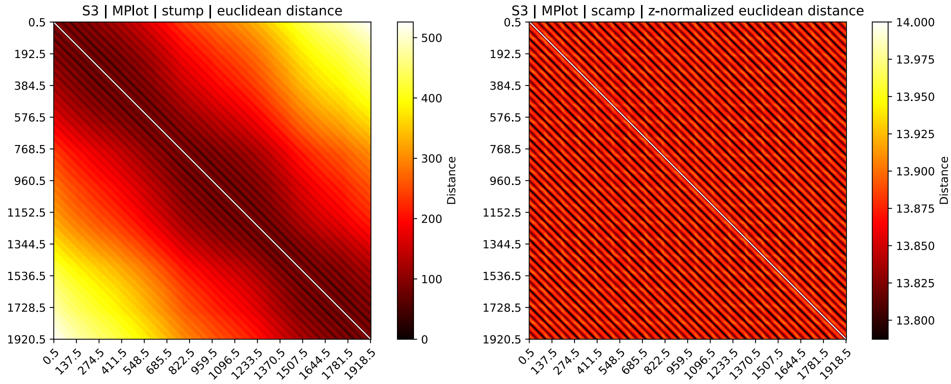

Fig. 16: S3 MPlot. On the left, STUMP with Euclidean distance is used. On the right, using SCAMP with the z-normalized euclidean distance.

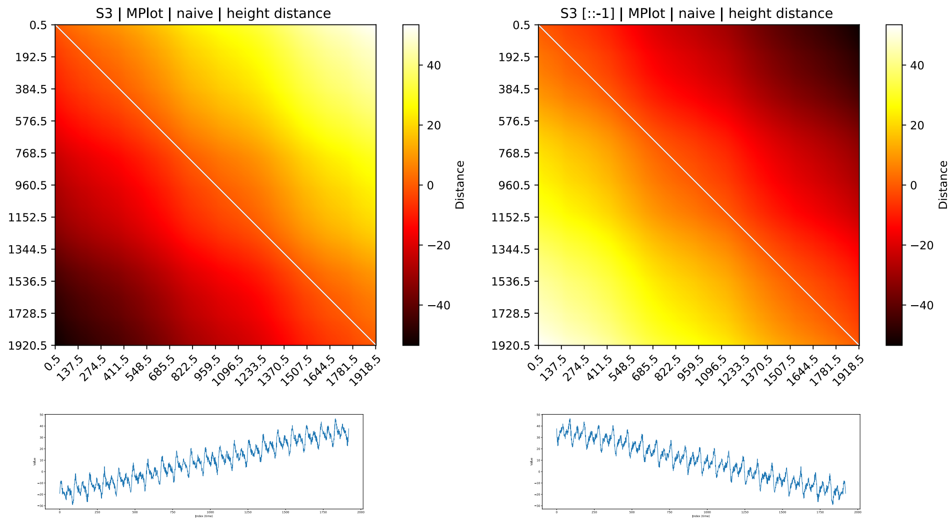

Fig. 17: S3 pseudo-MPlot using the proposed no-distance function to check a possible trend. By the left, the original time series is used, with an upward trend. By the right, the symmetric time series, with a downward trend.

are the following. The MPlot is a great way to analyze the time series data to find repeat patterns (heartbeat-SPO_2, see Fig. 7) and novelets (when SPO_2 is different from one moment and beyond). It is also ideal for detecting breakpoints in the time series: indexes $10K$, $12K$, and $17K$ in Fig. 19 where the plot looks "similar" until that moment and suddenly changes after that point.

However, it is more difficult to check in the MPlot the relation between the time series and the patterns and segment the different patterns that have been found. The interactiveness between the MPlot and the timeseries facilitates this proccess, but it is

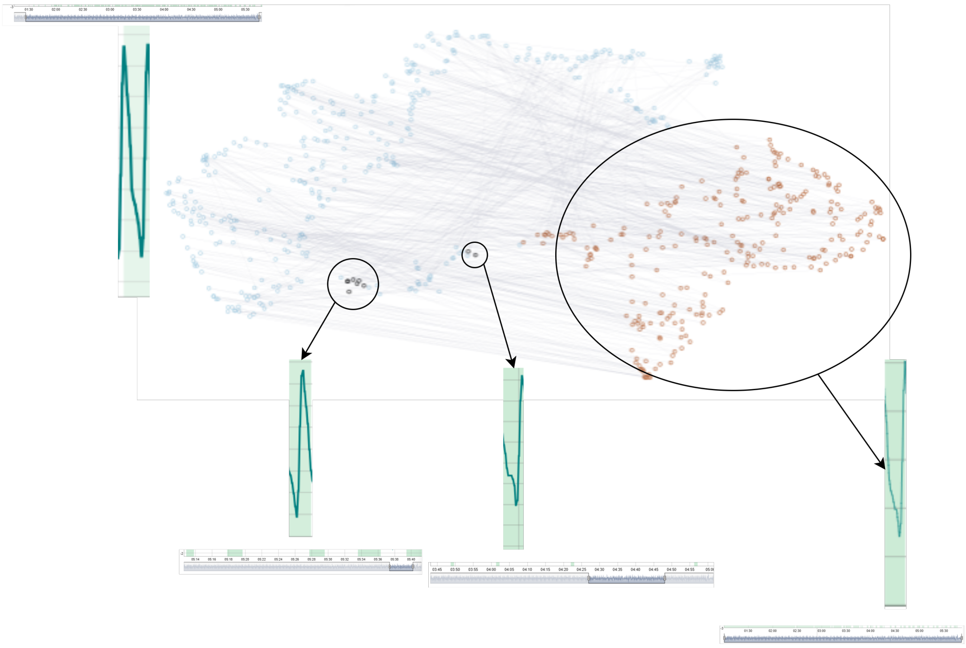

Fig. 18: Pulsus Paradoxus embedding space: checking patterns and anomalies.

still easier to visually segmentate the time series patterns by checking the clusters than going through the MPlot (see Fig. 18). Seasonalities and repetitive patterns are easy to check in both plots. Anomalies can be seen on both plots, but depending on the parameters it can be easier to see in the embedding space as an appart-point or in the MPlot as a white space. Altogether, they allow an outstanding analysis of univariate time series.

4 Conclusions and future work

DeepVATS has demonstrated excellent capabilities for the analysis of univariate and multivariate time series based on the visual interaction with the embedding space of Machine Learning models, more specifically, Deep Learning models. Mplots have been used in different areas like bioacustics, entomology or astronomy [23], were DeepVATS could also be utilized. By introducing MPlot in a more efficient and flexible version (including the specific function to measure trends), we can leverage DeepVATS in univariate time series analysis (see Fig. 20). Also, the integration with MPlot allows the user to have an easy interactive app that allows the interaction with the MPlot to visualize the relation with both the time series and the MatrixProfile. It also gives new powerful functionality for a fast preview of the behaviors of the time series, resulting in a plot obtained in less than 10 seconds (see Table 1), enhancing DeepVATS interactiveness. Although MPlot cannot be used for multivariate time series; the MatrixProfile has an analog definition for that case. The addittion of multidimensional MatrixProfile

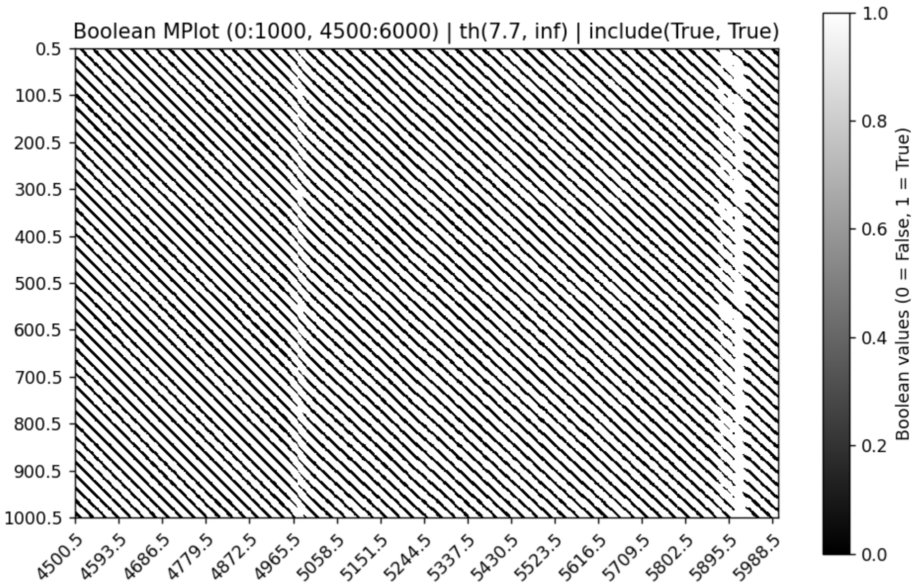

Fig. 19: MPlot computed using *scamp* and z-normalized euclidean distance, with $threshold = 7.7$. In the position $5k$ corresponding to $10k$, we observe a pattern similar to the one shown in Fig. 6. At position $5.8k$, another repetitive pattern is found.

is future work. Adding MatrixProfile-based features to the time series as another variable for model training could be interesting for concise analysis. Furthermore, some new foundation models [29, 46, 47] are appearing that can be useful to detect trends and could enhance the analysis within DeepVATS. Therefore, and to improve the development and functionalities of the DeepVATS tool, there are two lines of future work that we consider fundamental: checking the use of MPlot-derived features (such as the multidimensional Matrix Profile) for better analysis and the inclusion of new foundational models to the Deep Learning module.

Funding

This work has been supported by the next projects: H2020 TMA-MSCA-DN TUAI project “Towards an Understanding of Artificial Intelligence via a transparent, open and explainable perspective” (grant agreement no: 101168344); by Strategic Networking and Development Program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (RS-2023-00267476), and by EMIF managed by the Calouste Gulbenkian Foundation, in the project MuseAI.

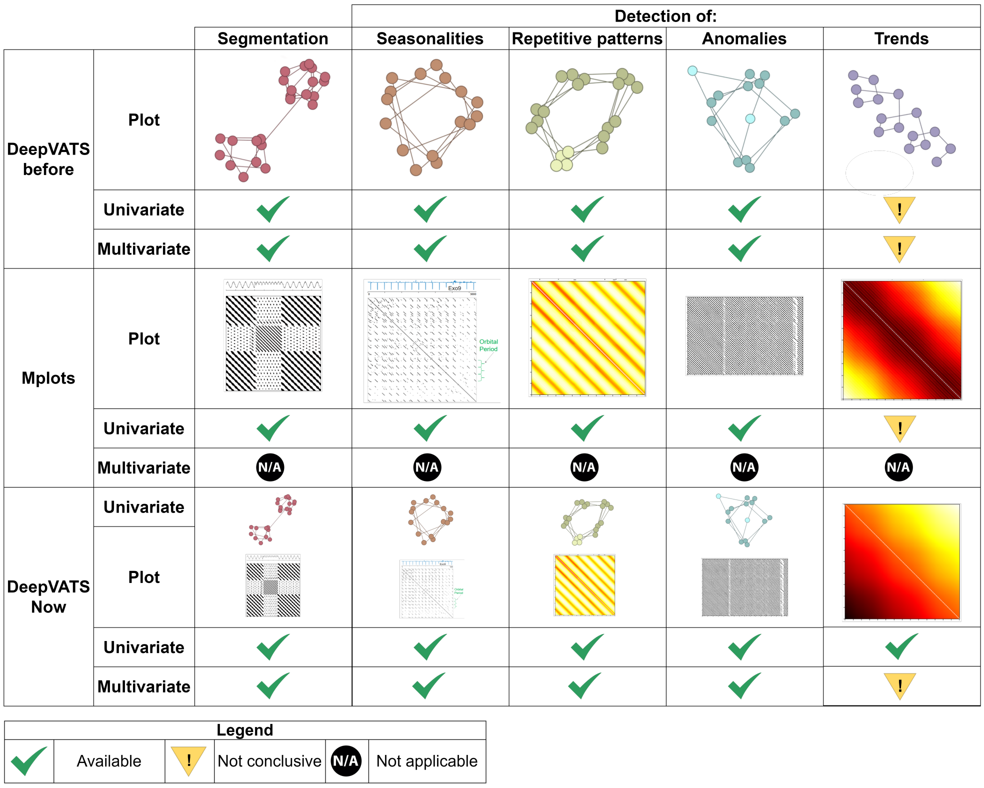

Fig. 20: Summary of DeepVATS capabilities before and after adding pseudo-MPlot. This Table has been adapted from [17, 24].

References

[1] Fernandes W, Komati KS, Assis de Souza Gazolli K. Anomaly detection in oil-producing wells: a comparative study of one-class classifiers in a multivariate time series dataset. Journal of Petroleum Exploration and Production Technology. 2024;14:343–363. https://doi.org/10.1007/s13202-023-01710-6.

[2] Liz-López H, Huertas-Tato J, Pérez-Aracil J, Casanova-Mateo C, Sanz-Justo J, Camacho D. Spain on Fire: A novel wildfire risk assessment model based on image satellite processing and atmospheric information. Knowledge-Based Systems. 2024;283:111198. https://doi.org/10.1016/j.knosys.2023.111198.

[3] Rodríguez-Fernández V, Menéndez HD, Camacho D. Analysing temporal performance profiles of UAV operators using time series clustering. Expert Systems with Applications. 2017;70:103–118. https://doi.org/10.1016/j.eswa.2016.10.044.

[4] Bello-Orgaz G, Mesas RM, Zarco C, Rodriguez V, Cordón O, Camacho D. Marketing analysis of wineries using social collective behavior from users' temporal activity on Twitter. Information Processing & Management. 2020;57(5):102220. https://doi.org/10.1016/j.ipm.2020.102220.

[5] Rama AM, Rodriguez-Fernandez V, Camacho D. Finding behavioural patterns among league of legends players through hidden markov models. In: EvoApplications 2020, Seville, Spain, April 15–17, 2020, Proceedings 23. Springer; 2020. p. 419–430. Available from: https://doi.org/10.1007/978-3-030-43722-0_27.

[6] Previti M, Rodriguez-Fernandez V, Camacho D, Carchiolo V, Malgeri M. Fake news detection using time series and user features classification. In: EvoApplications 2020, Seville, Spain, April 15–17, 2020, Proceedings 23. Springer; 2020. p. 339–353. Available from: https://doi.org/10.1007/978-3-030-43722-0_22.

[7] Villar-Rodríguez G, Souto-Rico M, Martín A. Virality, only the tip of the iceberg: ways of spread and interaction around COVID-19 misinformation in Twitter. Communication & Society. 2022;35(2):239–256. https://doi.org/10.15581/003.35.2.239-256.

[8] Wu R, Keogh EJ. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. IEEE Transactions on Knowledge and Data Engineering. 2023;35(3):2421–2429. https://doi.org/10.1109/TKDE.2021.3112126.

[9] Kang H, Kang P. Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism. Knowledge-Based Systems. 2024;290:111507. https://doi.org/10.1016/j.knosys.2024.111507.

[10] Blázquez-García A, Conde A, Mori U, Lozano JA. A Review on Outlier/Anomaly Detection in Time Series Data. ACM Comput Surv. 2021 apr;54(3). https://doi.org/10.1145/3444690.

[11] Gomez-Rosero S, Capretz MAM. Anomaly detection in time-series data using evolutionary neural architecture search with non-differentiable functions. Applied Soft Computing. 2024;155:111442. https://doi.org/10.1016/j.asoc.2024.111442.

[12] García R, Aguilar J. A meta-learning approach in a cattle weight identification system for anomaly detection. Computers and Electronics in Agriculture. 2024;217:108572. https://doi.org/10.1016/j.compag.2023.108572.

[13] Stevenson E, Rodriguez-Fernandez V, Minisci E, Camacho D. A deep learning approach to solar radio flux forecasting. Acta Astronautica. 2022;193:595–606. https://doi.org/10.1016/j.actaastro.2021.08.004.

[14] Schmidl S, Wenig P, Papenbrock T. Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment. 2022;15(9):1779–1797. https://doi.org/10.14778/3538598.3538602.

[15] McInnes L.: Interactive Visualizations. Accessed: 2024-06-26. Available from: https://umap-learn.readthedocs.io/en/latest/interactive_viz.html.

[16] TensorFlow.: Embedding projector - visualization of high-dimensional data. Accessed: 2024-07-02. Available from: https://projector.tensorflow.org/.

[17] Rodriguez-Fernandez V, Montalvo-Garcia D, Piccialli F, Nalepa GJ, Camacho D. DeepVATS: Deep visual analytics for time series. Knowledge-Based Systems. 2023;277:110793. https://doi.org/10.1016/j.knosys.2023.110793.

[18] Santamaria-Valenzuela I, Rodriguez-Fernandez V, Camacho D. Exploring Scalability in Large-Scale Time Series in DeepVATS framework. In: The 13th Conference on Information Technology and its Applications. Vietnam - Korea University; 2024. p. 1–12. Available from: https://arxiv.org/abs/2408.04692.

[19] Chen Z, Ma M, Li T, Wang H, Li C. Long sequence time-series forecasting with deep learning: A survey. Information Fusion. 2023;97:101819. https://doi.org/10.1016/j.inffus.2023.101819.

[20] Liang Y, Wen H, Nie Y, Jiang Y, Jin M, Song D, et al. Foundation models for time series analysis: A tutorial and survey. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024. p. 6555–6565. Available from: https://doi.org/10.48550/arXiv.2403.14735.

[21] Zimmerman Z.: SCAMP: The fastest way to compute matrix profiles on CPU and GPU. Accessed: 2024-06-26. https://github.com/zpzim/SCAMP.

[22] Zhu Y, Yeh CCM, Zimmerman Z, Kamgar K, Keogh E. Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds. In: ICDM 2018. IEEE; 2018. p. 837–846. Available from: https://doi.org/10.1109/ICDM.2018.00099.

[23] Shahcheraghi M, Mercer R, Rodrigues JMdA, Der A, Gamboa HFS, Zimmerman Z, et al. Introducing Mplots: scaling time series recurrence plots to massive datasets. Journal of Big Data. 2024;11(1):96. https://doi.org/10.1186/s40537-024-00954-1.

[24] Shahcheraghi M, Mercer R, De Almeida Rodrigues JM, Der A, Gamboa HFS, Zimmerman Z, et al. Matrix Profile XXVI: Mplots: Scaling Time Series Similarity Matrices to Massive Data. In: ICDM 2022. IEEE; 2022. p. 1179–1184. Available from: https://doi.org/10.1109/ICDM54844.2022.00151.

[25] Keogh E, Chakrabarti K, Pazzani M, Mehrotra S. Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems. 2001;3:263–286. https://doi.org/10.1007/PL00011669.

[26] Law SM. STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining. The Journal of Open Source Software. 2019;4(39):1504. https://doi.org/10.21105/joss.01504.

[27] Mgammal MH, Al-Matari EM, Alruwaili TF. Value-added-tax rate increases: A comparative study using difference-in-difference with an ARIMA modeling approach. Humanit Soc Sci Commun. 2023;10:121. https://doi.org/10.1057/s41599-023-01608-y.

[28] Chen Z, Zhou D, Zio E, Xia T, Pan E. Adaptive transfer learning for multimode process monitoring and unsupervised anomaly detection in steam turbines. Reliability Engineering & System Safety. 2023;234:109162. https://doi.org/10.1016/j.ress.2023.109162.

[29] Goswami M, Szafer K, Choudhry A, Cai Y, Li S, Dubrawski A. MOMENT: A Family of Open Time-series Foundation Models. arXiv preprint. 2024;https://doi.org/10.48550/arXiv.2402.03885.

[30] Mubarak H, Hammoudeh A, Ahmad S, Abdellatif A, Mekhilef S, Mokhlis H, et al. A hybrid machine learning method with explicit time encoding for improved Malaysian photovoltaic power prediction. Journal of Cleaner Production. 2023;382:134979. https://doi.org/10.1016/j.jclepro.2022.134979.

[31] Zhu Y, Zimmerman Z, Senobari NS, Yeh CCM, Funning G, Mueen A, et al. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE; 2016. p. 739–748. Available from: https://doi.org/10.1109/ICDM.2016.0085.

[32] Yeh CCM, Zhu Y, Ulanova L, Begum N, Ding Y, Dau HA, et al. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: ICDM 2016. IEEE; 2016. p. 1317–1322. Available from: https://doi.org/10.1109/ICDM.2016.0179.

[33] Nakamura T, Imamura M, Mercer R, Keogh E. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In: ICDM. IEEE; 2020. p. 1190–1195. Available from: https://doi.org/10.1109/ICDM50108.2020.00147.

[34] Lu Y, Srinivas TVA, Nakamura T, Imamura M, Keogh E. Matrix Profile XXX: MADRID: A Hyper-Anytime and Parameter-Free Algorithm to Find Time Series Anomalies of all Lengths. In: ICDM 2023. IEEE; 2023. p. 1199–1204. Available from: https://doi.org/10.1109/ICDM58522.2023.00148.

[35] STUMPY.: The Matrix Profile. Accessed: 2024-06-26. https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html#Time-Series-with-Length-n-=-13.

[36] Curtiss EI, Reddy PS, Uretsky BF, Cecchetti AA. Pulsus paradoxus: Definition and relation to the severity of cardiac tamponade. American Heart Journal. 1988;115(2):391–398. https://doi.org/https://doi.org/10.1016/0002-8703(88)90487-5.

[37] Shahcheraghi M, Mercer R, De Almeida Rodrigues JM, Der A, Gamboa HFS, Zimmerman Z, et al.: MPlot Catalog. Interpretation of MPlot's Patterns. Accessed 08/07/2024. Available from: https://docs.google.com/presentation/d/130jv-Ayd8pkc4aG52eUK5VbpI967np6x/edit?rtpof=true&sd=true.

[38] Biases W.: Experiment tracking with weights and biases. Available from: https://www.wandb.com.

[39] Oguiza I.: tsai: A state-of-the-art deep learning library for time series and sequences. GitHub. Accessed: 2024-07-02. https://github.com/timeseriesAI/tsai.

[40] McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint. 2018;https://doi.org/10.48550/arXiv.1802.03426.

[41] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11). www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?fbcl

[42] Scott BA, Johnstone MN, Szewczyk P, Richardson S. Matrix Profile data mining for BGP anomaly detection. Computer Networks. 2024;242:110257. https://doi.org/10.1016/j.comnet.2024.110257.

[43] Shabikay Senobari N, Shearer PM, Funning GJ, Zimmerman Z, Zhu Y, Brisk P, et al. The matrix profile in seismology: Template matching of everything with everything. Journal of Geophysical Research: Solid Earth. 2024;129(2):e2023JB027122. https://doi.org/10.1029/2023JB027122.

[44] Ghazvini MB, Sànchez-Marrè M, Naderi D, Angulo C. Anomaly Detection in Gas Turbines Using Outlet Energy Analysis with Cluster-Based Matrix Profile. Energies. 2024;17(3):653. https://doi.org/10.3390/en17030653.

[45] Godahewa R, Bergmeir C, Webb G, Abolghasemi M, Hyndman R, Montero-Manso P.: Solar Power Dataset (4 Seconds Observations). Zenodo. Available from: https://zenodo.org/records/4656027.

[46] Ansari AF, Stella L, Turkmen C, Zhang X, Mercado P, Shen H, et al. Chronos: Learning the language of time series. arXiv preprint. 2024;https://doi.org/arXiv:2403.07815.

[47] Woo G, Liu C, Kumar A, Xiong C, Savarese S, Sahoo D. Unified training of universal time series forecasting transformers. arXiv preprint. 2024;https://doi.org/10.48550/arXiv.2402.02592.