# Dual-Metric Clustering for Multivariate Time Series: KMeans with DTW and QuadTree with Entropy*

**Samuel R. Torres**[1], **Raphael Saldanha**[3], **Rocío Zorrilla**[1], **Victor Ribeiro**[1], **Eduardo H. M. Pena**[2], **Fábio Porto**[1]

National Laboratory of Scientific Computing (LNCC)
Caixa Postal 25651-075 – Petrópolis - RJ - Brazil

Federal University of Technology - Paraná,
Campos Mourão - PR - Brazil

Institut national de recherche en sciences et technologies du numérique (INRIA)

{samuelrt, victorr}@posgrad.lncc.br, {romizc, fporto}@lncc.br

eduardopena@utfpr.edu.br, raphael.de-freitas-saldanha@inria.fr

***Abstract.*** *The efficacy of machine learning models are contingent on input data quality and model selection itself. In this work we highlight the importance of data quality, particularly in identifying regions within the input space that exhibit similar behavior. Clustering is used to group similar data, and is explored for their potential to enhance model performance by identifying these regions. The aim of this paper is to provide insights into the effectiveness of using clustering to improve machine learning model performance.*

## 1. Introduction

This research focuses on developing and evaluating clustering techniques to identify regions with similar behaviors in spatiotemporal datasets. Our methodology involves creating these clusters, or regions, using K-Means with Dynamic Time Warping (DTW) and Quadtree algorithms. K-Means-DTW groups time series data by aligning sequences of different lengths, while Quadtree partitions spatial data into regions. By reducing entropy and enhancing pattern recognition, these clusters enable us to develop both global models, which capture broad trends, and local models, which are fine-tuned to specific regional variations.

While initially applied to meteorological data, this approach is versatile and can be used for detecting extreme events and other applications. The results demonstrate the effectiveness of these clustering methods in forming coherent regions, which will be further validated through model training and performance evaluation. This research not only enhances clustering techniques but also provides a robust framework for future studies in various fields requiring precise pattern detection.

## 2. Related Works

This section reviews the literature relevant to our research on time series clustering. This work compares two clustering approaches for multivariate time series to improve the per-

*The authors acknowledge the Brazilian funding agencies CNPq, CAPES and Petrobras Termo de Cooperação 0050.0122040.22.9 for their financial support in the development of this work.

formance of global and local models. We highlight key methodologies and findings, identify gaps in the literature, and explain how our research contributes to this area.

[Montero-Manso and Hyndman 2021] et al. highlight three principles for forecasting groups of time series: adding features, increasing memory, and partitioning. They describe partitioning as splitting the set into individual series for a local approach, while the global approach works on the trivial partition that keeps the set intact. [Vázquez et al. 2021] et al. review techniques for clustering Multivariate Time Series (MTS) data. It starts with APCA-MINDIST, which represents variables by segments using Adaptive Piecewise Constant Approximation (APCA) and measures distances with MINDIST. APCA-DTW follows a similar approach but uses DTW. FFT-hclust applies Fast Fourier Transform (FFT) to z-scored data, retains the top 10 components, and uses energy differences to measure distances. Finally, CMD-hclust combines a compression-based dissimilarity measure (CMD) with hierarchical clustering, employing a sliding window approach for time series of different lengths. All methods use hierarchical clustering and the "elbow rule" to determine the number of clusters. A novel methodology for forecasting univariate and multivariate time series is presented in [Castán-Lascorz et al. 2022]. This methodology involves segmenting the time series into windows of equal length and grouping them into $k$ clusters, denoted as $G_1, \ldots, G_k$, based on their characteristics.

Clustering time series enables the identification of dataset subsets whose instances share similar data distribution. In [Ribeiro et al. 2023], the authors propose and evaluate the training of subset models for each data subset and compare their performance against a global model, which is a model built on the full dataset. Nevertheless, our objective is to present that alternative partitioning methods can yield favorable outcomes under controlled input entropy conditions, as elucidated in this study. This approach has the potential to enhance model efficacy and optimize training duration, thereby contributing to advancements in computational efficiency and model performance evaluation.

## 3. Background

This section explores two strategies for partitioning and grouping multivariate time series data: k–Means clustering with DTW and Quad-Tree partitioning based on spatial coordinates and entropy. The k–Means clustering is a widely used method for partitioning data into distinct groups. However, for multivariate time series data, traditional metrics like Euclidean distance often fall short due to the temporal dependencies and varying lengths inherent in time series data. To address this, DTW is employed as the dissimilarity measure [Warren Liao 2005]. DTW aligns time series sequences by warping the time axis to minimize the distance between them, effectively capturing temporal distortions and similarities. The k–Means algorithm, using DTW, iteratively assigns each time series to the nearest cluster centroid, recalculates the centroids based on these assignments, and repeats the process until convergence [Cormen et al. 2022] . Furthermore, the traditional k–Means algorithm is not designed to handle time series data directly due to temporal variability and differing series lengths. However we have the Dynamic Time Warping (DTW), DTW is an algorithm for measuring similarity between two time series which may vary (i.e.warp) in timing [Mueen and Keogh 2016]. The underlying mathematics of DTW involves computing the minimum distance alignment between two sequences, accommodating local temporal distortions by optimizing the cumulative similarity measure.

Let $x = x_1, \ldots, x_n$ and $y = y_1, \ldots, y_m$ be two time series. We build an $n \times m$ grid where each index $(i, j)$ represents the value of a metric $\delta(i, j)$ between points $x_i$ and $y_j$. A path $\mathrm{W} = w_1, \ldots, w_k$ defines the DTW distance between $x$ and $y$, as follows:

$$\mathrm{DTW}(x, y) = \min_{w} \left[ \sum_{k=1}^{p} \delta(w_k) \right] \tag{1}$$

The second strategy employs a Quadtree for spatial partitioning, using latitude and longitude, and integrating entropy to assess dissimilarities. By splitting regions with the highest entropy, the Quadtree method effectively partitions the input space into areas of relatively homogeneous time series. This ensures that the resulting groups are geographically coherent and similar in their time series characteristics.

The quad tree is a data structure appropriate for storing information to be retrieved on composite keys [Finkel and Bentley 1974]. Quadtrees can be used to store different types of data. We will describe the variant that stores a set of points in the plane. In this case the recursive splitting of squares continues as long as there is more than one point in a square. So the definition of a Quadtree for a set $P$ of points inside a square $\sigma$ is as follows. Let $\sigma \coloneqq [x_\sigma : x'_\sigma] \times [y_\sigma : y'_\sigma]$.

- If $card(P) \leq 1$ then the Quadtree consists of a single leaf where the set $P$ and the square $\sigma$ are stored.
- Otherwise, let $\sigma_{NE}, \sigma_{NW}, \sigma_{SW}$, and $\sigma_{SE}$ denote the four quadrants of $\sigma$. Let $x_{mid} \coloneqq (x_\sigma + x'_\sigma)/2$ and $y_{mid} \coloneqq (y_\sigma + y'_\sigma)/2$, and define:

$$\begin{aligned} P_{NE} &\coloneqq \{p \in P : p_x > x_{mid} \quad \text{and} \quad p_y > y_{mid}\} \\ P_{NW} &\coloneqq \{p \in P : p_x \leq x_{mid} \quad \text{and} \quad p_y > y_{mid}\} \\ P_{SW} &\coloneqq \{p \in P : p_x \leq x_{mid} \quad \text{and} \quad p_y \leq y_{mid}\} \\ P_{SE} &\coloneqq \{p \in P : p_x > x_{mid} \quad \text{and} \quad p_y \leq y_{mid}\} \end{aligned}$$

The choice of using less-than-or-equal-to and greater-than in the definition of the sets $P_{NE}, P_{NW}, P_{SW}$, and $P_{SE}$ means that we define the vertical splitting line to belong to the left two quadrants, and the horizontal splitting line to the lower two quadrants[de Berg et al. 2008].

## 4. Methodology

Let $D$ be a dataset such that $D = \{a_1, \ldots, a_n\}$. Consider a partitioning method $P$ that partitions $D$ into $P(D) = \{p_1, \ldots, p_m\}$ such that $p_1 \cup \ldots \cup p_m = D$. Let entropy$(D)$ denote the entropy of the dataset $D$. It is assumed that there exists a partitioning $P''$ of $D$ such that entropy$(D) \geq$ entropy$(P''(D))$. The problem is to define $P''$. In this work, we consider two algorithms for $P''$: K-Means clustering and Quadtree.

First, select a dataset $D$ and compute its initial entropy. Next, determine the optimal number of clusters $k$ and apply k-means clustering to $D$, ensuring that the resulting clusters reduce the entropy. Then, apply Quadtree partitioning to $D$, with constraints on the size and minimum entropy of each partition $p_i$. Finally, compare the entropy results from the k-means and Quadtree methods, and select the method that yields the lowest entropy for $D$. This approach ensures that the selected partitioning method effectively reduces the entropy of the dataset, resulting in more homogeneous regions.

## 5. Experiments and Results

In this section, we present our results on the partitioning of input data and the identification of regions exhibiting similar behavior. For this experiment, we collected data from 37 meteorological stations in Rio de Janeiro from 2016 to 2023. However, for our specific objectives, we utilized data from a single year, spanning from January 1, 2021, to December 31, 2021. The variables analyzed include latitude, longitude, precipitation, humidity, temperature, and wind speed. Although the data were originally recorded hourly, we aggregated the values over 24-hour periods for our analysis. For each meteorological station, the information is consolidated into a multivariate time series with 365 time steps, corresponding to daily measurements. To identify regions with similar behavior, we will utilize the two primary algorithms described in the previous section.

The computational execution was conducted on a Dell PowerEdge R730 machine, equipped with 768GB of RAM, dual Intel(R) Xeon(R) CPU E5-2690 v3 processors operating at 2.60GHz, and a Tesla P100 GPU with 16GB of VRAM.

### 5.1. Results by k–Means clustering with DTW

In the current experiment, latitude and longitude are not included as variables for analysis. Their inclusion could potentially bias the clustering of time series solely based on geographical proximity, disregarding other informative variables. To explore various clustering scenarios, we will conduct nine executions, each varying the number of clusters from two to ten. To evaluate the effectiveness of this clustering, we use the silhouette score. This score measures how well each data point fits into its assigned cluster compared to neighboring clusters, ranging from $[-1, 1]$. A higher silhouette score indicates that clusters are well-separated and data points are appropriately grouped.

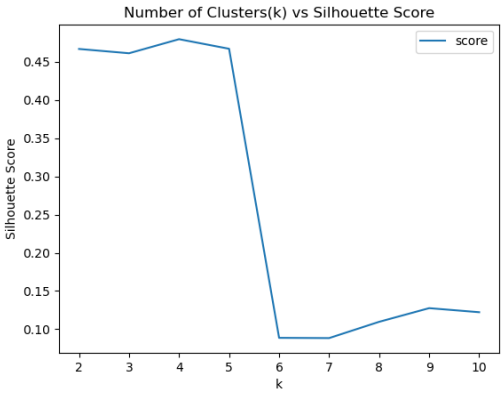


**Figure 1. k–Means Performance with DTW Across Varying Cluster Counts**

In this instance, the quality of clustering is assessed using the silhouette score. As depicted in Figure 1, our results indicate that the optimal clustering occurs when the number of clusters is four. This result signifies that we have identified four distinct regions where meteorological variables exhibit similar patterns, independent of geographic locations. Essentially, we are discussing how these regions are delineated based on meteorological similarities rather than geographical proximity. It is also important to note that the number of clusters obtained using the k–Means method tends to be less than or

equal to the number of clusters obtained using the Quadtree method. This is expected because the k–Means algorithm has the flexibility to allocate points with similar behavior more effectively. Unlike Quadtree, k–Means does not have spatial boundaries, which allows it to group similar points that may be in different spatial regions. Additionally, the computational cost in terms of execution time for this experiment was approximately two minutes, compared to the subsequent experiment, this is relatively quick.

### 5.2. Results by Quadtree based on entropy

In the current experiment, latitude and longitude are employed to create a grid for the Quadtree algorithm, which identifies spatiotemporal regions where entropy is low or partially low. This involves setting an entropy threshold that specifies regions with equal to or less than the admitted entropy level. Entropy will be measured using a pairwise matrix based on DTW for each point within the target region. On the other hand, it is possible that nearby points may exhibit similar behavior or not.

The primary objective of the experiment is to ensure that during the search process, if the algorithm encounters a region with entropy exceeding the predefined threshold, it will subdivide the region into four smaller regions. This process will continue iteratively until the entropy of the resulting regions is lower than the threshold and the number of points within each region is below the minimum required for forming a cluster.

The following results demonstrate how the algorithm effectively reduces input entropy in many cases. We created a grid showing the accepted entropy levels and the minimum number of points required for forming a cluster. We will present the extreme cases, which we define as particularly demanding scenarios: one where the entropy threshold is 0.1 and the minimum number of elements is two, and another where the entropy threshold is 0.7 and the minimum number of elements is six.

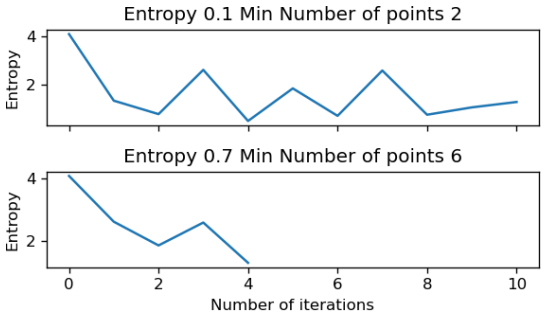


**Figure 2. The number of iterations required for the search to conclude**

Figure 2 illustrates the varying rates of entropy reduction. It is important to note that the iterations required for the search to conclude have an inverse relationship: as the entropy threshold and the minimum number of elements permitted to form a cluster increase, the number of iterations decreases. For a clearer illustration of how the algorithm divides the space.

In the Figure 3, each point represents a meteorological station. It is noteworthy that the result obtained in this instance is ten clusters, with an entropy threshold of 0.1 and two minimum number of points. However, evaluating the quality of these clusters

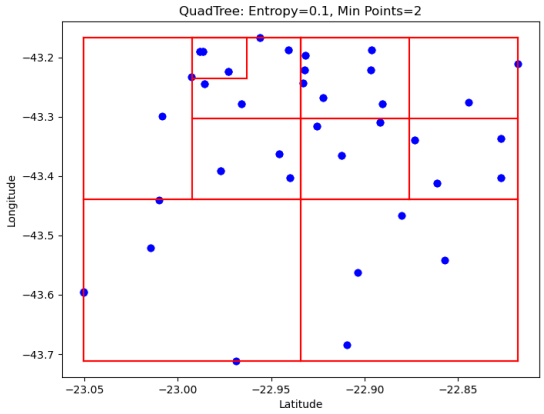


**Figure 3. The input space divided by the Quadtree algorithm.**

directly is challenging due to the absence of a metric like silhouette score. Additionally, running this experiment nine times significantly increases the computational cost, taking nearly 32 minutes to complete. However, one of our objectives is to measure the quality of these clusters. To achieve this, we need to train models and evaluate their performance. We expect the local models to perform similarly or better than the global model, but this analysis will be revisited in subsequent work.

## 6. Conclusions and Future Works

Our experiments demonstrate that both methods for identifying regions with similar behavior within the input space effectively reduce input entropy. The K-Means algorithm, which can form clusters independently of spatial information and evaluate them using silhouette scores. The computational cost for K-Means clustering is relatively low, with execution times for our experiments of approximately two minutes for each run. Conversely, the Quadtree method delineates regions where spatial proximity plays a crucial role in the analysis. This approach highlights instances where points nearby may exhibit disparate behaviors, underscoring the importance of spatial context alongside temporal information. However, the computational cost for the Quadtree method is significantly higher, with experiments taking nearly 32 minutes to complete. Overall, while K-Means offers a quicker and computationally efficient solution, the Quadtree method provides deeper insights into spatial-temporal dynamics, particularly in regions with high entropy. The choice between these methods depends on the specific requirements of the analysis, balancing the trade-offs between computational efficiency and the depth of spatial-temporal.

As future works, we are working towards the following objectives: initially, reducing entropy by creating clusters using both k-Means with DTW and Quadtree methods, which delineate regions with similar behavior, already completed. Additionally, we plan to improve the steps involved in calculating distances as they are calculated inside the Quadtree, as our results indicate this is crucial for optimizing the execution time of the Quadtree. Subsequently, we will develop global and local models based on the clusters derived from the initial step to evaluate their predictive performance.

## References

Castán-Lascorz, M., Jiménez-Herrera, P., Troncoso, A., and Asencio-Cortés, G. (2022). A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting. *Information Sciences*, 586:611–627.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to algorithms*. MIT press.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg.

Finkel, R. and Bentley, J. (1974). Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9.

Montero-Manso, P. and Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, 37(4):1632–1653.

Mueen, A. and Keogh, E. J. (2016). Extracting optimal performance from dynamic time warping. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2129–2130. ACM.

Ribeiro, V., Pena, E. H. M., de Freitas Saldanha, R., Akbarinia, R., Valduriez, P., Khan, F. A., Stoyanovich, J., and Porto, F. (2023). Subset modelling: A domain partitioning strategy for data-efficient machine-learning. In *Proceedings of the 38th Brazilian Symposium on Databases, SBBD 2023, Belo Horizonte, MG, Brazil, September 25-29, 2023*, pages 318–323. SBC.

Vázquez, I., Villar, J. R., Sedano, J., and Simić, S. (2021). A comparison of multivariate time series clustering methods. In *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020) 15*, pages 571–579. Springer.

Warren Liao, T. (2005). Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874.