



Detection of anomalous vehicular traffic and sensor failures using data clustering techniques

Davide Moretti^{1,2} · Elia Onofri^{1,3} · Emiliano Cristiani¹

Received: 1 April 2025 / Accepted: 3 July 2025
© The Author(s) 2025

Abstract

The increasing availability of traffic data from sensor networks has created new opportunities for understanding vehicular dynamics and identifying anomalies. In this study, we employ clustering techniques to analyse traffic flow data with the dual objective of uncovering meaningful traffic patterns and detecting anomalies, including sensor failures and irregular congestion events. We explore multiple clustering approaches, i.e. partitioning and hierarchical methods, combined with various time series representations and similarity measures. Our methodology is applied to real-world data from highway sensors, enabling us to assess the impact of different clustering frameworks on traffic pattern recognition. We also introduce a clustering-driven anomaly detection methodology that identifies deviations from expected traffic behaviour based on distance-based anomaly scores. Results indicate that hierarchical clustering with symbolic representations provides robust segmentation of traffic patterns, while partitioning methods such as k -means and fuzzy c -means yield meaningful results when paired with Dynamic Time Warping. The proposed anomaly detection strategy successfully identifies sensor malfunctions and abnormal traffic conditions with minimal false positives, demonstrating its practical utility for real-time monitoring. Real-world vehicular traffic data are provided by Autostrade Alto Adriatico S.p.A.

Keywords Traffic data clustering · Time series analysis · Anomaly and sensor failure detection · Intelligent transportation systems

Mathematics Subject Classification 90B20 · 62M10

1 Introduction

The rapid proliferation of sensors in modern technology has led to an unprecedented surge in data generation. From industrial monitoring systems to wearable devices and smart

cities, sensors continuously collect vast amounts of information. These data streams are inherently temporal and often unstructured and unlabelled, requiring specialised techniques to extract meaningful insights. The intricate structure of time series often conceals valuable patterns and relationships from superficial analysis, hence requiring advanced data mining techniques. Consequently, time series data mining has found applications in multiple disciplines for tasks such as anomaly detection, motif and pattern discovery, indexing, clustering, classification, visualisation, trend analysis, and forecasting [1].

Among these tasks, *clustering* plays a crucial role in uncovering latent structures within unlabelled temporal datasets. It involves partitioning data into homogeneous groups based on a similarity measure that quantifies how “close” the data objects are in a given space. The choice of a suitable (set of) distance function(s) is often application dependent, and selecting the appropriate one, along with a representation space, is a challenge in itself. Furthermore,

✉ Davide Moretti
davide.moretti@uniroma3.it

Elia Onofri
elia.onofri@cnr.it

Emiliano Cristiani
e.cristiani@iac.cnr.it

¹ IAC Istituto per le Applicazioni del Calcolo, CNR National Research Council of Italy, Via dei Taurini, 19, 00185 Rome, RM, Italy

² Department of Mathematics and Physics, University of Roma Tre, L. S. L. Murialdo, 1, 00146 Rome, RM, Italy

³ IAC Istituto per le Applicazioni del Calcolo, CNR National Research Council of Italy, Via Pietro Castellino 111, 80131 Naples, NA, Italy

datasets suitable for data mining techniques are usually large and high-dimensional, demanding significant computational resources.

A notable example of large-scale time series data generation comes from traffic sensors deployed on highways. Such sensors continuously monitor and record vehicle flow passing beneath them, capturing key attributes such as vehicle counting and corresponding speeds as well as grouping vehicles by class (e.g. light and heavy vehicles). These pieces of information are crucial for traffic management, congestion prediction, and road safety analysis and are typically used in traffic model calibration and validation. However, the data collected by these sensors are inherently temporal, vast in size, high-dimensional, and often unlabelled, making it challenging to analyse using traditional methods. For instance, even distinguishing between an empty road and a fully congested one—both scenarios in which no vehicles pass the sensor—presents an interesting challenge that heavily relies on the temporal nature of data [2].

In this context, clustering techniques can help uncover patterns in traffic behaviour, such as recurring congestion events, seasonal variations, or anomalous fluctuations that might indicate accidents or roadwork. While extensive research has been devoted to applying clustering in traffic analysis, little work has been done on anomaly detection; most studies assume that sensors operate flawlessly and do not account for the possibility of sensor failures in their models. However, real-world traffic monitoring systems are prone to a wide range of sensor malfunctions that can severely impact data reliability. These failures manifest in various forms, some of which are not immediately evident. Common issues include missing data, incorrect temporal aggregation (e.g., misaligned timestamps), abnormally large spikes in recorded values (caused e.g. by sudden fluctuation on the sensor power line), or even data streams that remain constant over time despite expected variations. Such anomalies can introduce significant biases in traffic models and, if undetected, lead to inaccurate conclusions in downstream analyses.

Detecting sensor failures is particularly challenging because erroneous data may still appear plausible when considered on its own. For instance, a sensor might report a realistic number of passing vehicles or average speeds, yet these values could be inconsistent w.r.t. historical trends or be in conflict with data from neighbouring sensors. Given the diversity of potential failure modes, it is unreasonable to assume that all possible faults can be identified a priori. Instead, automated approaches capable of dynamically detecting anomalies are needed. By revealing unexpected deviations in traffic patterns, clustering techniques offer a promising solution.

The use of clustering for anomaly detection, however, comes with various challenges. Clustering results still require expert validation before being operationalised, as repeated anomalies can potentially accumulate into a distinct, well-

formed cluster. This phenomenon is particularly problematic in clustering methods that require a predefined number of clusters; if a method is expected to identify k typical clusters but instead detects \hat{k} anomalous clusters, then the number of discernible normal behaviours is effectively reduced to $k - \hat{k}$. Consequently, some meaningful traffic patterns may go undetected. This issue becomes even more critical when anomalies evolve into persistent trends, making it difficult to distinguish between transient faults and fundamental shifts in sensor behaviour (e.g. a sensor whose miscalibration goes unnoticed). Addressing these challenges requires ad hoc strategies to ensure that clustering techniques remain robust in dynamic environments where anomalies might transition into new steady-state behaviours.

1.1 Relevant literature

Clustering

Clustering is a fundamental unsupervised learning technique that partitions data into groups, or clusters, based on some notion of similarity measure. Classical clustering algorithms can be broadly categorised into partition-based, hierarchical, density-based, and model-based approaches [3, 4]. Among partition-based methods, k -means [5] and fuzzy c -means [6] remain two of the most widely used approaches due to their efficiency and simplicity, particularly when applied to time series data [3]. However, they suffer from limitations such as assuming spherical clusters (for some space embedding) and requiring the number of clusters to be specified in advance. Hierarchical clustering methods, like, e.g., agglomerative clustering, allow for a flexible exploration of cluster granularity without requiring a predefined number of clusters [7]. Density-based approaches, such as DBSCAN [8], are particularly useful in detecting clusters of arbitrary shape and are well suited for handling noise and outliers. On the other hand, model-based techniques, such as Gaussian Mixture Models (GMMs) [9], assume an underlying probabilistic structure and provide soft clustering assignments, making them particularly effective when clusters exhibit overlapping distributions. Due to their often streaming characteristics, time series data can also benefit from a dynamic clustering approach [10]. Dynamic clustering adapts the clustering outcome based on the continuous data flow, either by updating an object's cluster membership over time or by modifying cluster definitions altogether in response to novel data.

Despite significant advancements in clustering methodologies, their effectiveness heavily depends on selecting an appropriate similarity measure, which must be coupled with a suitable embedding space to represent the data. In traditional clustering applications where data points reside in Euclidean spaces, the L_2 norm provides an intuitive, robust, and reliable similarity measure. However, in more complex settings, particularly with time series data, defining meaningful distances

becomes a significant challenge. Time series are inherently sequential, high-dimensional, and prone to temporal distortions, necessitating the development of specialised similarity measures and embedding techniques [4].

Time series clustering

Existing approaches for time series clustering can be divided into three main categories:

Whole time series clustering, which treats each time series as a single entity and applies classical similarity measures such as the Euclidean distance (easier yet unable to capture temporal distortions) or more advanced metrics, e.g., the Dynamic Time Warping (DTW), originally introduced as a speech pattern recognition technique in [11] and later applied in many other fields [12].

Subsequence clustering, which segments time series into smaller subsequences before clustering. This approach is useful for detecting repeated patterns, such as motifs in physiological or financial data, but can suffer from spurious patterns due to overlap [13].

Feature-based clustering, which transforms time series into a lower-dimensional space using statistical descriptors [14], wavelet coefficients [3], or shapelet-based representations [15], improving efficiency and interpretability while reducing computational complexity.

The latter approach is particularly useful when dealing with large-scale datasets, where direct similarity evaluations are computationally prohibitive, as it happens in our case study. Among the simplest feature-based techniques, Piecewise Aggregate Approximation (PAA) [16] provides an efficient way to reduce dimensionality by summarising segments of time series with average values. This technique has been successfully applied in numerous domains and later combined within Piecewise Dynamic Time Warping (PDTW) [17], an adaptation of DTW designed for compressed representations [18].

Building on these ideas, symbolic representations have gained traction for their efficiency in handling large-scale datasets. Symbolic Aggregate approxImation (SAX) [19] discretises continuous values into symbolic categories based on a Gaussian-spaced grid, naturally coupled with MINDIST, a tailored distance metric confronting symbols on the underlying grid [20]. A large amount of work was performed in this direction, introducing numerous variants including the Extended SAX (ESAX) [21], which retains also information about minima and maxima of the interval, the Indexable SAX (iSAX) [22], particularly suited for terabyte-sized time series, and the SAX-trends [23], incorporating trend information to enhance clustering accuracy.

Evaluating clustering performance

When it comes to assessing clustering kindness, many different methodologies and indicators can be employed.

With some labelled examples provided, classical solutions like precision, accuracy, and recall can be used, potentially paired with the analysis of the corresponding ROC curves. However, when no information is available on correct data classification, a solution that takes into account intra- and inter-cluster discrepancies must be used. Among the mostly used, we recall the *Silhouette index*, originally introduced in 1987 in [24] as an indicator to distinguish between well- and badly separated clusters, later adopted also for non-Euclidean metrics, see e.g. [25]. Also worth of mention is the *Partition Coefficient and Exponential Separation* (PCAES) which, originally introduced in [26], is specifically useful in the context of fuzzy clustering, i.e. when points are not assigned to a specific cluster only, but rather the method provides the probability of a given point being in any specific cluster. For a general treatise on the usage and comparison of cluster validity indices, we refer the interested reader to [27].

Additionally, hierarchical clustering methods provide insights into clustering robustness by analysing the stability of significant clusters across different levels. For instance, in [28], the author explored how monitoring the number of p -significant clusters can reveal patterns in data segmentation.

Anomaly detection

Anomaly detection is a fundamental task in data analysis, aiming to identify unusual patterns, rare events, or deviations from expected behaviour [29].

Among various approaches to anomaly detection, clustering-based methods provide a natural way to distinguish between normal and abnormal instances. Here, clusters represent groups of typical patterns, while points deviating significantly from their nearest clusters are flagged as anomalies. The distance from the closest cluster centroid is commonly used as an anomaly score, often referred to as *reconstruction error* [30]. This method is particularly effective in unsupervised settings, where anomalies are unknown a priori and must be inferred from the structure of the data.

Beyond centroid-based techniques, alternative strategies have been developed to improve robustness, including (i) Density-Based Anomaly Detection, (ii) Reconstruction-Based Approaches, and (iii) Time Series Anomaly Detection. Within the latter category, methods such as Local Outlier Factor (LOF) [31] and DBSCAN-based outlier scoring [8] identify anomalies by analysing local density variations. Conversely, techniques such as autoencoders [32] and Principal Component Analysis (PCA)-based reconstruction errors [33] have been used to detect anomalies as deviations from learned representations. Finally, in particular for sequential data, distance measures such as Dynamic Time Warping (DTW) anomaly scores [34] or forecasting residuals from Long Short-Term Memory (LSTM) networks [35] highlight unexpected temporal variations.

In what follows, we adopt a clustering-driven anomaly detection approach, using distance-based anomaly scoring to measure deviations from expected patterns.

Traffic analysis

Managing, understanding, and predicting traffic is a fundamental aspect of transportation planning, infrastructure development, and smart city initiatives [36]. Traffic data are inherently temporal, comprising information such as vehicle count, speed, congestion levels, and travel times, often collected across multiple locations in a road network (spatial component). Extracting meaningful insights from these data is crucial for optimising traffic flow, reducing congestion, improving road safety, and supporting intelligent transportation systems (ITS) [37].

In recent years, the increasing availability of high-resolution traffic data from sensors, IoT devices, and connected vehicles has led to a paradigm shift in traffic analysis. Alongside, GPS data also proved to be useful (provided a sufficient number of samples is considered); noteworthy examples include, e.g., [38], where the authors apply dynamic clustering to traffic data to analyse real-time urban traffic and correctly identify congestions, later refined in [39].

In [40], the authors show how traditional methods, which adopt aggregated and low-frequency data, are progressively being replaced by real-time monitoring and forecasting techniques that leverage machine learning and artificial intelligence.

While real-time forecasting focuses on short-term traffic dynamics, offline approaches remain crucial for long-term traffic pattern analysis, strategic infrastructure planning, and anomaly detection. Early approaches include, for example, works like [41], where the authors analysed historical highway traffic flow data using clustering to distinguish common patterns repeating across different locations due to daily and seasonal features. In particular, the approach of analysing flux data within the various clustering frameworks also received considerable attention within the recent literature. It is the case, e.g., of [42] where subsequences of traffic data are clustered performing an embedding to reduce the dimensionality and where DTW is used to compare the features vectors. A similarity heatmap is then created between sensors to reconstruct spatial similarity from the temporal clusters. In a similar fashion, Toshniwal et al. [43] try to capture the underlying traffic pattern and phenomena by spatially and temporally clustering flux data gathered in Aarhus (Denmark) while trying to correlate different behaviours with different street types. Also worth attention is the work [44], where an analysis of urban traffic data is performed, offering a comparison between different possible clustering approaches on the same dataset. Here, both internal validation indices and macroscopic pieces of information about roads, vehicles, and people are applied to evaluate and validate the results.

However, despite this progression, time series clustering application in identifying traffic anomalies—such as sensor malfunctions or unexpected disruptions—has received less attention. Sensor failures in traffic monitoring systems can lead to inaccurate data collection, ultimately affecting downstream analytics and decision-making. Developing automated, clustering-based approaches for detecting anomalous traffic sensor behaviour could provide a robust, data-driven solution for ensuring data integrity in large-scale traffic monitoring networks.

1.2 Contributions

In the present work, we present a comprehensive study on clustering-based traffic analysis, focusing on the identification of typical traffic patterns and anomaly detection using real-world highway sensor data. The key contributions of this study are as follows:

Comparison of clustering techniques. We conduct an extensive evaluation of various clustering methodologies, including partitioning (k -means, fuzzy c -means) and hierarchical approaches, applied to time series traffic data. Our analysis highlights the impact of different time series representations and similarity measures on clustering effectiveness.

Clustering-driven anomaly detection. We propose a novel approach that integrates clustering results with anomaly detection mechanisms. By analysing deviations from cluster centroids, our method effectively identifies irregular traffic patterns, with a particular focus on sensor malfunctions and traffic flux anomalies.

Multivariate time series analysis. We extend the clustering framework to a multivariate setting, incorporating data from multiple sensor sources and multiple vehicle classes aggregation. This approach improves the detection of complex anomalies that are not apparent in univariate analyses.

Real-world application. Our methodologies are applied to a dataset of traffic data collected by highway sensors dispatched in northern Italy, prompting insights into real traffic conditions. We also develop an automated pipeline for traffic pattern analysis and anomaly detection, providing a practical tool for practitioners to monitor and ensure the integrity of traffic data.

1.3 Paper organisation

The rest of the paper is organised as follows. In Sect. 2, we introduce the dataset used in the study, including data sources, preprocessing steps, and initial univariate (Sect. 2.1) as well as multivariate (Sect. 2.2) analyses. In Sect. 3, we present the clustering methodologies, time series rep-

representations, similarity measures, and evaluation criteria adopted in our study (Sect. 3.1). We also describe the novel anomaly detection framework based on the clustering results (Sect. 3.2). In Sect. 4, we discuss the findings of our clustering experiments (Sect. 4.1), compare different methodologies, and evaluate the performance of our anomaly detection system (Sect. 4.2). Representative case studies of detected anomalies are also discussed. Finally, in Sect. 5, we summarise the key findings of the study, discuss its implications, and outline potential directions for future research. The paper concludes with Appendix A, where we discuss a previously unpublished approach to averaging symbolic series.

2 Discussing the real-world data

The dataset used in this paper is provided by Autostrade Alto Adriatico S.p.A. (previously Autovie Venete S.p.A.), which manages various highway segments in northern Italy, namely the A4 Venezia–Trieste, A23 Palmanova–Udine Sud, A28 Portogruaro–Conegliano, A34 Villesse–Gorizia and part of the A57 Tangenziale di Mestre, for a total road length of roughly 234 km.

The traffic flow in the aforementioned area is monitored via a large number of cameras, mobile sensors, and fixed sensors. Fixed sensors are gathered into *stations*, which are placed along the highway in fixed positions. Each station is associated with a single traffic direction, and the distance between them is not constant, ranging from 2 to 20 km. Sensors in different stations can be based on different technologies, but they perform in a standardised way: each sensor in each station monitors *a single lane*, providing traffic data every minute, which includes flux (number of vehicles passed under the sensor), average velocity, and occupancy rate. The relative error on counting and velocity claimed by the manufacturer is of $\pm 3\%$.

In this paper, we only consider flux data, which are further divided on the basis of the class of vehicle, following the German TLS 5+1 class standard.¹ We aggregate classes 1 and 2 in a new macro-class named *light vehicles* and classes 3, 4, and 5 in another macro-class named *heavy vehicles*. An aggregate class 0 is also provided, obtained by summing the total vehicular flux in the time interval and performing a weighted average for the speed and occupancy rates.

2.1 Univariate analysis

For the basic analysis, we focus on aggregated flux data in class 0. Our dataset is composed of all the daily time series of flux data measured by every sensor and spanning the entire

year of 2022. The “atomic datum” of our analysis is, therefore, a time series of length 1440 (one flux datum for every minute of the day). Figure 1 shows visual examples of “normal” data, i.e. data expected for a usual day with no sensor anomalies, no relevant accidents, and no traffic congestion; a smoothed version of the data obtained via convolution with a Gaussian kernel is also depicted in the figure for better readability.

Along with regular data, it is important to note that daily time series might present different anomalies, some of which are worth further investigation during the dataset construction and whose removal would yield clearer results.

One significant example is represented by the presence of an abnormal number of minutes in which flux zero is recorded. Such a situation, see Fig. 2(top left), is typically hard to explain without contextual information, like e.g. the presence of a construction site or some kind of lane closure; hence, we decided to remove time series presenting a total flux not exceeding a given small threshold.

A different example is given by a time series with missing flux data. This outcome might be related to a sensor failure, a connection failure, or a database storage failure. Although in some cases realistic values might be inferred, such a procedure is not completely trustworthy and might not be applicable if gaps are long in time, see e.g. Fig. 2(bottom left). Hence, we decided to remove any time series with any missing data to form the cleaned dataset we later used for clustering analysis.

Finally, it is worth noting that traffic anomalies might also appear. However, differently from the previous two cases, no simple method for automatic detection can be mounted. As an example, do consider that a flux reading of zero could be transmitted by the sensor due to two different opposing situations: (i) the road is empty and no vehicle passed under the sensor or (ii) vehicles are completely stopped and no one pass through the sensor due to a full congestion (queue); similar outcomes can be derived for contexts with low fluxes and discriminating the two cases is an interesting problem *per se*, also when velocities are considered, as discussed in [2]. Figure 2(right) depicts two different traffic anomalies where both these situations can be spotted: at night, we have some null flux data due to the empty road, while during the day, we have some null flux data caused by full congestions. Due to the intrinsic difficulties in detecting them, such anomalies are among the most interesting ones in our scenario and, although we removed some of them via heuristic and manual analysis, a consistent number is still present in the cleaned dataset.

A final cleaned dataset comprising 31,125 daily time series forms the base of our univariate analysis.

¹ <https://www.bast.de/DE/Publikationen/Regelwerke/Verkehrstechnik/Unterseiten/V5-tls.html>.

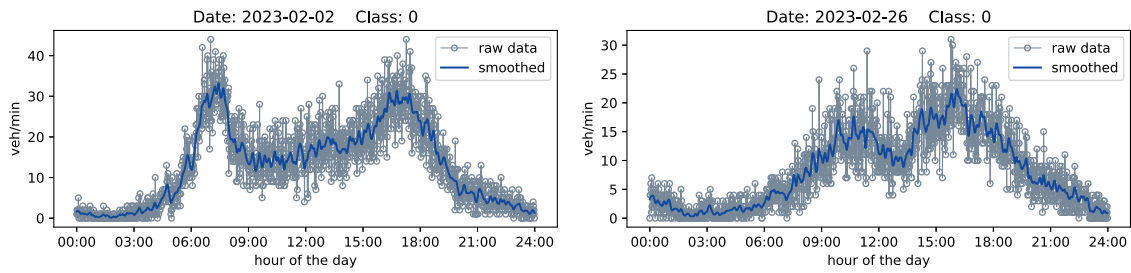


Fig. 1 Examples of normal single-day time series. The characteristic double-hump behaviour of flux data can be seen in different time frames. Do notice that, for the sake of readability, y-axes are not homogeneous

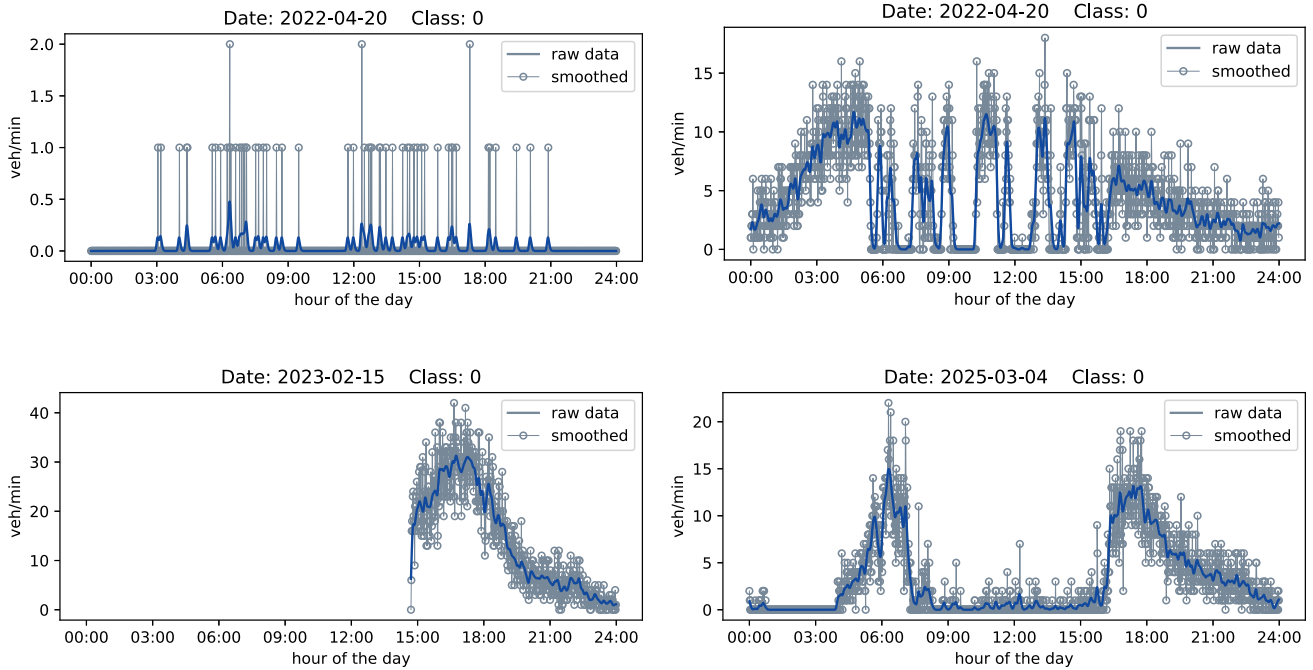


Fig. 2 Examples of anomalous one-day behaviour. On the left, probable sensor failures that are automatically removed from the dataset. On the right, probable traffic anomalies, which are harder to detect automatically. Do notice that, for the sake of readability, y-axes are not homogeneous

2.2 Multivariate analysis

For a more advanced and detailed analysis, we also consider finer data coming from multiple sensors simultaneously, yet belonging to the same station and the same day. The “atomic datum” is, therefore, the combination of two or more daily time series derived from sensor data after a partial lane and/or class aggregation. Among various attempts we explored in our analysis, we here report the three most promising types of multivariate data aggregations, a summary of which is found in Table 1. In particular, we considered heavy vehicles on the sole driving (slow) lane as *Variable A*: here, it is important to recall that heavy vehicles are not theoretically allowed on the passing (fast) lane. Then, we paired *Variable A* with (T1) light vehicles in the same lane, (T2) light vehicles in the same station, and (T3) light vehicles divided between the driving

Table 1 Different approaches for the multivariate analysis

Type	Variable A	Variable B	Variable C
T1	Driving lane	Driving lane	–
	Heavy vehicles	Light vehicles	
T2	Driving lane	All lanes	–
	Heavy vehicles	Light vehicles	
T3	Driving lane	Driving lane	Passing lane
	Heavy vehicles	Light vehicles	Light vehicles

and passing lanes. Figure 3 depicts a typical behaviour of each approach.

For this analysis, we have considered a small dataset made of 158 days from a single station. Despite the small size of the dataset, it allows us to provide a proof of concept of how a higher level of detail actually improves traffic data analysis.

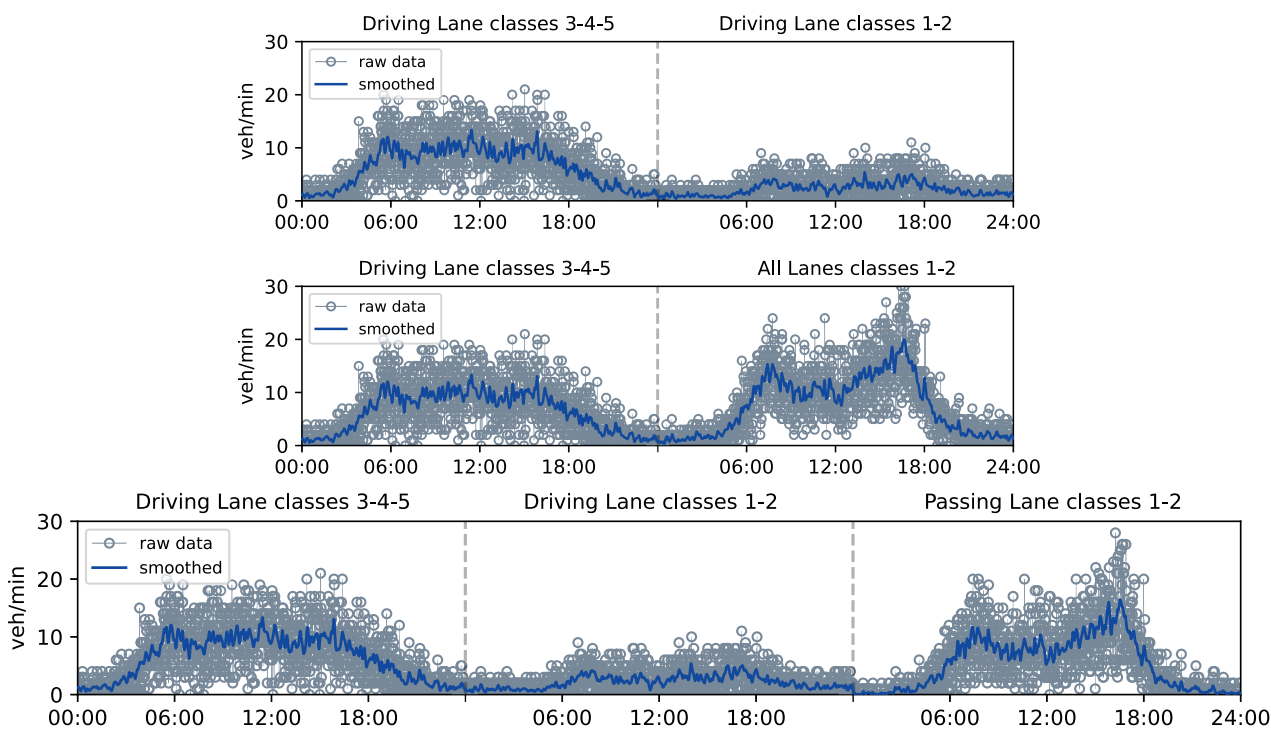


Fig. 3 Example of normal single-day multivariate time series for the three types of analysis introduced in Table 1 (T1–T3 from top to bottom). Each box within a panel corresponds to one variable

3 Methods

In order to provide a clear understanding of the proposed analysis, in this section we present an overview of the implemented methods, which are later compared. The confident reader can safely skip to Sect. 3.2, where our novel approach to anomaly detection is presented.

3.1 Clustering

In what follows, we focus specifically on PAA, SAX, ESAX, MINDIST, DTW, *k*-means+, fuzzy *c*-means+, agglomerative hierarchical clustering, and clustering result evaluation.

Time series representation: PAA

It is one of the simplest time series representations, capable of dimensionality reduction while still matching the original shape of the series. A time series $s = \{x_1 \dots x_n\} \in \mathcal{S}$ of length n can be represented in a w -dimensional space \mathcal{S}' (with $w \leq n$) as a vector $\bar{s} = \{\bar{x}_1 \dots \bar{x}_w\}$ with the i -th element of the sequence being

$$\bar{x}_i = \left\lceil \frac{w}{n} \right\rceil \sum_{j=\lfloor \frac{n}{w}(i-1) \rfloor + 1}^{\lceil \frac{n}{w} i \rceil} x_j, \tag{1}$$

where the value $c = \frac{n}{w}$ represents the amplitude of the averaging window and is often referred to as compression rate.

Time series representation: SAX

It is a symbolic representation of time series based on PAA. The general idea of SAX is to transform the piecewise approximated series \bar{x} into a “word” \hat{x} composed of symbols from some ℓ -length alphabet $\mathcal{A} = \{\alpha_1, \dots, \alpha_\ell\}$, where each symbol α_i is present with comparable frequency among all the words of the dataset. To do so, the dataset is normalised and PAA-ed. Obtained values are assumed to be distributed according to some chosen distribution, with a Gaussian $\mathcal{N}(0, 1)$ being the typical choice. A set of $\ell + 1$ breakpoints $-\infty = \beta_0 < \beta_1 < \dots < \beta_{\ell-1} < \beta_\ell = \infty$ are then built such that they equally split the chosen distribution in segments of area $1/\ell$, i.e. such that $\int_{\beta_i}^{\beta_{i+1}} \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) dx = 1/\ell$ when dealing with $\mathcal{N}(0, 1)$. The PAA approximated values \bar{x}_i are then converted as

$$\hat{x}_i = \alpha_j \quad \text{if} \quad \beta_{j-1} \leq \bar{x}_i < \beta_j. \tag{2}$$

Time series representation: ESAX

It represents an extension of the regular SAX where the PAA applied over the normalised dataset retains the minimum and maximum value of each time frame, along with the regular average defined as in (1). Resulting values are then converted according to (2) in a 3ℓ -length word, where each triplet of letters represents minimum, average, and maximum of the time frame.

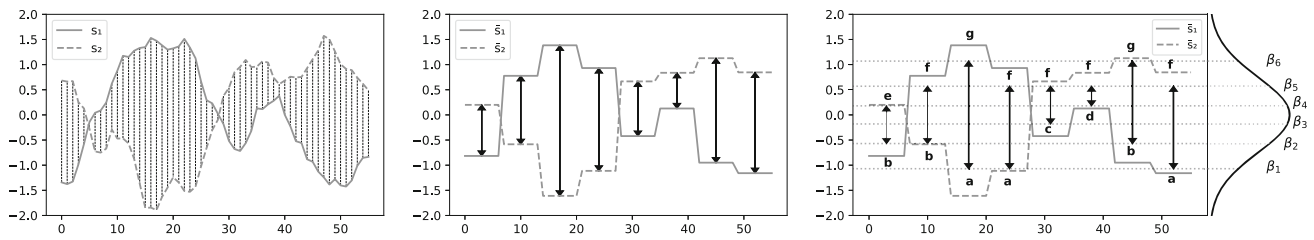


Fig. 4 A visual depiction of the considered distances on (left) L_2 over normalised time series, (centre) L_2 over PAA representation, (right) MINDIST over SAX representation, when applied on the same couple of series s_1 and s_2

Similarity measure: Euclidean norm

Given two time series $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$, the classical L_2 norm

$$L_2(A, B) = \sum_{i=1}^n \sqrt{a_i^2 - b_i^2} \quad (3)$$

can be applied to any numeric time series offering a competitive non-elastic measure of similarity (see also Fig. 4(left) and (centre)). In particular, L_2 norm applied on PAA-ed series and on raw series is comparable if rescaled by the size of the time frame.

Similarity measure: MINDIST

It is a point-wise symmetric symbolic similarity measure defined on symbolic series originally introduced along with SAX representation (Fig. 4(right)). It assesses similarities between two characters α_i and α_j as the smallest distance between the two corresponding intervals, namely

$$\text{dist}(\alpha_i, \alpha_j) = \begin{cases} \beta_{j-1} - \beta_i, & \text{if } i < j - 1, \\ 0, & \text{if } |i - j| \leq 1, \\ \beta_{i-1} - \beta_j, & \text{if } i > j + 1. \end{cases} \quad (4)$$

Similarity measure: DTW

It is a nonlinear metric originally proposed in the field of speech recognition and widely adopted in the literature on time series clustering. Its widespread use is comparable to that of the L_2 norm, but it is usually more accurate, especially when applied to small datasets [3]. In fact, in the DTW, each point of the first time series is compared to an arbitrary point of the second one while maintaining the respective ordering, hence resulting in a virtual stretching/compression of the original series (see also Fig. 5).

More formally, let $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_m\}$ be two series (of potentially different length [45]), and let δ be some distance measure defined over the corresponding value domain. Then DTW (in its iterative definition) evaluates the best alignment, finding a warping path π [46, 47] by progressively building the subsequences distance matrix and finding the path of least cost. Multiple formulations and algorithms have been proposed in the literature over time,

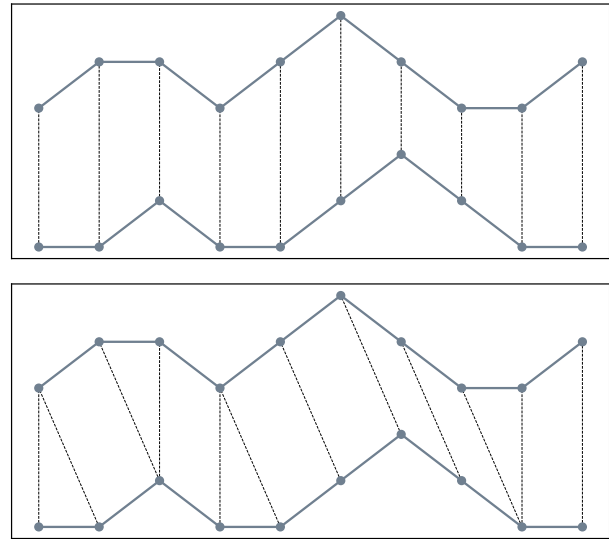


Fig. 5 Visual representation on the different comparisons applied by the non-elastic Euclidean distance (top) and elastic DTW (bottom) over a mock time series

offering slight variations and improvements, including normalisation options and rootless versions; in what follows, we adopt the formulation from [1, 46], i.e.

$$\text{DTW}(A, B) = \min_{\pi} \left\{ \sqrt{\sum_{(i,j) \in \pi} (a_i - b_j)^2} \right\}. \quad (5)$$

The choice is motivated by the fact that (3) can be seen as a special case of (5), obtained when the best alignment is point-wise. This makes the two distances intrinsically comparable with one another. DTW is also a computationally intensive metric to evaluate; hence, in our application, we limit the possible warping path spreading using the Sakoe–Chiba band [1, 11].

Clustering method: k-means

Being one of the most common clustering approaches, k -means regroups the objects in clusters around k centroids, which are obtained, in turn, as the mean of the clusters themselves. k -means is then an iterative deterministic process which converges to a local optimum that depends solely

on the initial centroids. Consequently, different ways of determining the initial conditions were developed, the most adopted being k -means+ [48], which progressively assigns initial centroid centres according to probability distributions over the datapoints themselves.

In what follows, centroid evaluation is carried out with unweighted mean on regular and PAA-ed time series, Dynamic Barycenter Averaging (DBA [47]) on DTW-ed series, and a custom median-based algorithm built on the MINDIST for symbolic series; the latter approach is, to the best of our knowledge, novel and a complete description is found in Appendix A.

Clustering method: fuzzy c -means

It can be seen as a variation in regular k -means, where cluster membership is somewhat replaced with a notion of affinity, meaning that in fuzzy c -means, each data point belongs to a specific cluster with some probability depending on the distance from the cluster centroid. To alleviate the typical convergence issues which can be found in high-dimensional spaces [49], we adopt the fuzzy c -means+ initialisation algorithm [50] with fuzziness coefficient $m = 2$.

Clustering method: agglomerative hierarchical clustering analysis (HCA)

It iteratively builds a hierarchy of the dataset in a bottom-up fashion, starting with a cluster per element and pair-wise merging progressively the clusters containing close elements. In our experiments, element distance is extended to cluster distance according to the average linkage strategy. To extract the clusters from the so-formed hierarchy (i.e. cut the dendrogram), we employed a heuristic originally introduced in [28], seeking plateaux in the number of p -significant clusters to find a stable phase of the clustering process. If no such phase exists, the clustering is deemed a failure.

Clustering evaluation

To evaluate the results of k -means+ to the dataset, we chose the number of clusters based on the Silhouette index of the clustering result [24], a value in $[-1, 1]$ obtained as the average difference of inter-cluster dissimilarity and intra-cluster dissimilarity. Conversely, to evaluate the results of the fuzzy c -means+, we adopted the PCAES [26], consisting of a combination in $[-k, k]$ between a normalised partition coefficient of the fuzzy partition matrix and an exponential separation measure hindering clusters too close to one another. While a normalisation in $[-1, 1]$ seems natural, it is not advised [26]. Finally, for what concerns HCA, we selected the valuable outputs under the stability of the p -significant clusters.

As a final note, it is important to recall that some approaches required hyper-parameters to be fixed a priori (or tuned). A summary of the five main approaches considered, along with the corresponding hyper-parameter choices, is reported in Table 2. Here, do mind that DTW is not directly used as PDTW is adopted instead. This is due to the pro-

hibitive dimensions involved (dataset size and series length), which would yield prohibitive computational costs.

Going multivariate

All the discussed representations, methods, and indices can be naturally extended to multivariate series using different approaches. In particular, our multivariate data are made of a small number of features only (either two or three features, cf. Table 1), so we decided to process the features independently for what concerns representation and evaluation. Similarity scores are hence evaluated on the single sub-series and then summed together.

In the following discussion, we present the PDTW approach only as a proof of concept for the multivariate series, being the most promising among the five.

3.2 Anomaly detection

In this section, we detail our approach to anomaly detection by building over the various clustering techniques introduced so far. In particular, given the daily nature of our data, and provided we would like to create a methodology running on a daily basis to detect and communicate anomalous behaviour, let \mathcal{X}_d be the set of time series collected in a given day d , purged from trivial anomalies like missing data (cf. Sect. 2.1). Then, the idea is to create a suitable set of ℓ anomaly scores (one per each clustering considered) that can be used to assess how a given time series $s \in \mathcal{X}_d$ deviates from the corresponding clusterings and process such scores to build a set $\mathcal{A}_d \subset \mathcal{X}_d$ of the “potential anomalous” data.

In detail, for each considered hard clustering (k -means, HCA), the distance from the closest centroid typically provides a good proxy of the abnormality of the datum, hence forming a simple yet effective anomaly score. Conversely, when considering soft clustering (fuzzy c -means), we use the classical approach consisting of averaging the distance from each centroid weighted on the probability of belonging to the corresponding cluster.

The resulting set of ℓ anomaly scores can then be collected in a ℓ -rows $|\mathcal{X}_d|$ -columns matrix $A = (a_{i,j})$ and processed via the following two aggregated scores:

Aggregated Average (AGG), which row-rescales the matrix A in the interval $[0, 1]$ (i.e. apply the $(0, 1)$ -normalisation on each score) and averages the resulting scores (i.e. perform the average-by-column). In other words, the aggregated score for the j -th series is obtained as

$$\text{AGG}(s_j) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{a_{i,j}}{\max\{a_{i,\cdot}\}} \quad (6)$$

Table 2 Summary of representation and chosen distance measures

Name	Representation	Similarity	Hyperparameters
E	Raw data	Euclidean	–
PAAE	PAA	Euclidean	$w = 144$
SAX	SAX	MINDIST	$w = 144, a = 9$
ESAX	ESAX	MINDIST	$w = 144, a = 9$
PDTW	PAA	DTW	$w = 144, sakoe-chiba$ radius = 6

forming an ‘average distance’ in $[0, 1]$ from the ‘normality’ (the higher, the more anomalous).

Positional Ranking (POS), which applies the rank-transformation by row on A (i.e. maps score in $[1, \dots, |\mathcal{X}_d|]$ from the higher to the lower value) before evaluating the AGG as before. In other words, POS rescale scores in $\{0, 1/|\mathcal{X}_d| - 1, \dots, |\mathcal{X}_d| - 2/|\mathcal{X}_d| - 1, 1\}$ before evaluating an average score, hence providing an anomalous ranking in $[0, 1]$ (the lower, the more anomalous).

Classically, these scores can be then used to discriminate anomalies depending on some fixed thresholds. However, it is important to note that the normalisation is performed day-wise, i.e. for each daily dataset \mathcal{X}_d , different values $\max\{a_{i, \cdot}\}$ and $\min\{a_{i, \cdot}\}$ are chosen; this is due to the online nature of the process, where data from the future days are not available, making it impossible to define a fixed minimum or maximum. As a consequence, series from different days are not directly comparable, making it difficult to define a single threshold value.

Hence, to finally process a single series s , we follow the same rationale of POS, adopting a ranking approach which is, in principle, agnostic of anomalies. Let $2k$ be the number of potential anomalies to report daily (in our experiments $k = 3$, $2k \sim 5/100|\mathcal{X}_d|$). Then, two initial sets of daily anomalies are formed as the most relevant k anomalies according to AGG and POS (i.e., the k top scoring for AGG and k bottom scoring for POS), which we will call TOPAGG and TOPPOS, respectively.

Exploiting the intrinsic temporal feature of the datasets \mathcal{X}_d , we can further analyse \mathcal{Y}_d in terms of the anomalies detected in the previous days $\mathcal{Y}_{d-1}, \dots, \mathcal{Y}_{d-h}$, for some $h \in \mathbb{N}$. In fact, it is reasonable that sensor anomalies might repeat within different days, making them more worthy of attention. To combine this piece of information with the intrinsic rank provided by AGG and POS, we define the final confidence score $\mathfrak{c}(s)$ for a series $s \in \mathcal{X}_d$ in $[0, 3k]$ as the sum of the two ranks in TOPAGG and TOPPOS (if present, in inverted order from k to 1) plus k if $s \in \mathcal{Y}_{d-t}$, $t \in [1, h]$, for at least $g \leq h$ days.

The confidence score \mathfrak{c} can then be either used *as-is* or it can be classified in grades of confidence/severity. In our experiments, e.g., we classified anomalies as *mild* when

$\mathfrak{c} \in [1, k]$, *moderate* when $\mathfrak{c} \in (k, 2k]$, and *severe* when $\mathfrak{c} \in (2k, 3k]$. As a reference, we also set $h = 6$ and $g = 2$, meaning that *severe* anomalies can occur only when an anomaly is detected at least three times within the current week.

The overall methodology is summarised in Fig. 6.

4 Results

In this section, we first present some results about the clustering of the traffic data introduced in Sect. 2, followed by an analysis of traffic anomalies and sensor failure detection. Due to the size of the original dataset and the unclear definition of anomaly, no thorough attempt was made at creating a labelled dataset. Hence, no direct ground truth comparison is available to validate the presented results. During a six-month trial phase, results were validated through expert knowledge from stakeholders at Autostrade Alto Adriatico S.p.A., with the anomaly detection technique successfully identifying real anomalies consistently. The consistency of clustering indicators also provides a reliable proxy for assessing the effectiveness of different methodologies.

4.1 Clustering results

We present the clustering results divided according to the approach for the sake of readability.

k-means

Having no information *a priori* on the correct number k of clusters, we tried to identify a suitable value by making use of the Silhouette index. As depicted in Fig. 7 (left), the index shows a descending trend whichever representation is chosen (but the PDTW), prompting the less the cluster number the better performance. While the Silhouette score is not a perfect indicator, this suggests that *k*-means is not able to separate the dataset effectively.

In general, by observing the results, elements in the dataset seem to lack a clear separation in different groups, with data evenly distributed in the space (although with different densities). This, combined with the dataset dimension, motivates the reduced effectiveness of the *k*-means approach, which aims for a clear-cut separation of the data.

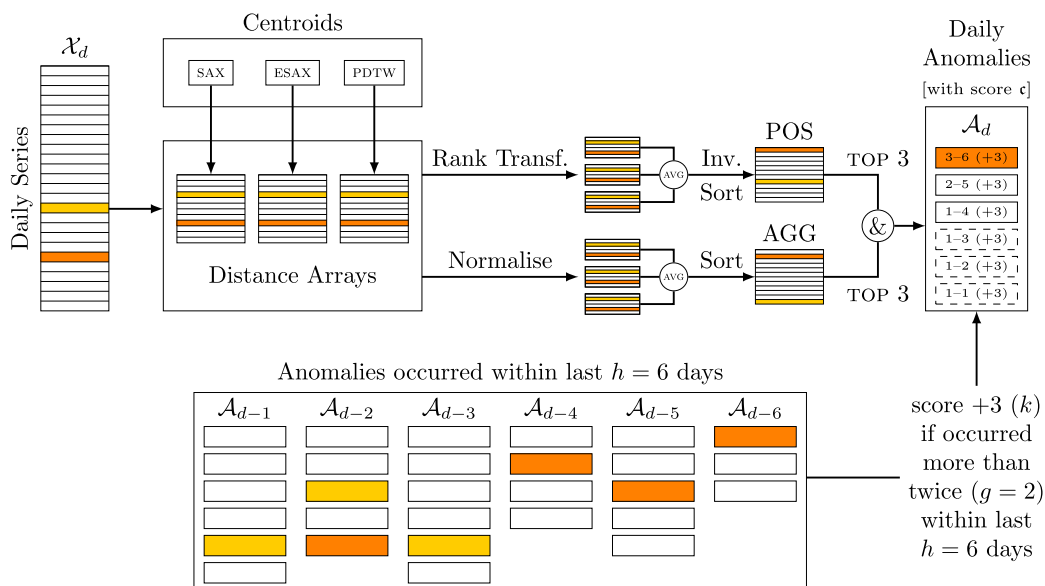


Fig. 6 Anomaly detection pipeline processing the daily series \mathcal{X}_d : the distance from the closest cluster centroid is evaluated for each clustering method and values per series $s \in \mathcal{X}_d$ are averaged after positional ranking (POS) and (0, 1)-Normalisation (AGG, cf. (6)); top $k = 3$ anomalies from both POS and AGG are considered, with a score $c(s) \in [1, 3k]$ given by their respective rank $(1-k)$ plus k whether the anomaly is recur-

ring $g = 2$ times within the last $h = 6$ days. $c(s)$ range for each anomaly is provided in the \mathcal{A}_d box, where dashed series are not guaranteed to be present. Two sample series are depicted in orange and yellow throughout the pipeline: red series is flagged as anomaly, being ranked top for POS, second top for AGG, and recurrent ($c = 8$); yellow series is not flagged as anomaly despite being recurrent

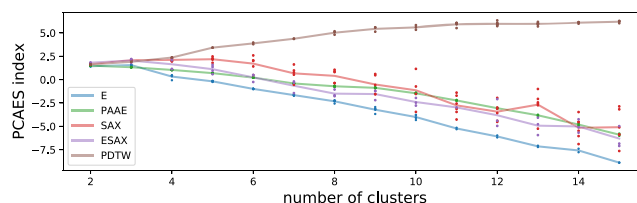
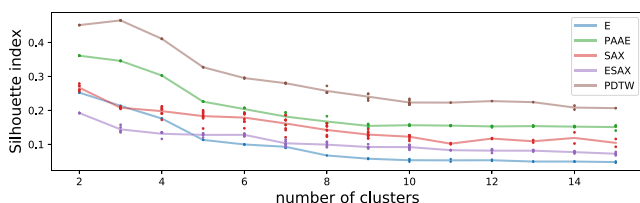


Fig. 7 Plot of the Silhouette index for k -means (left) and PCAES index for fuzzy c -means (right) applied to the 5 representation–similarity couples (cf. Table 2)

Fuzzy c -means

Analogous to k -means, we adopted PCAES to identify a suitable number c of clusters. As reported in Fig. 7 (right), most of the approaches present low PCAES values, which decrease to negative values when c increases, deeming their ineffectiveness. Conversely, PDTW representation offers promising results with monotone increasing values of PCAES. In particular, $c = 15$ represents a sweet-point after which the index reaches a plateau (values not shown for conciseness). Figure 8 shows the 15 resulting clusters along with the corresponding centroids. We can see how the division is mostly driven by the series “height”, but the high number of clusters helps distinguish different “shapes” sharing similar averages.

It is worth noting that in both k -means and fuzzy c -means the series “height” (average flux) seems to affect the clustering results predominantly, in particular if compared to the actual series “shape”. This is particularly evident in the best clustering of k -means (PDTW, $k = 3$, cf. Fig. 7 left, clusters

not shown for conciseness), where three clusters are formed collecting three orders of magnitude of the flux. While this does not represent a wrong result *per se*, it highlights how the distance selection can affect the direction the analysis takes.

Similar considerations can be drawn for the pairing of representation space and similarity score. In fact, the suboptimal results achieved by SAX and ESAX are not surprising in the context of partitioning methods: symbolic representation discretises the space, hence hampering the ability of k -means and fuzzy c -means to slowly move the cluster centres in the space until a suitable position is found, to the point that some initial parameter configurations failed to converge.

Hierarchical methods

As per the previous methods, we applied the hierarchical approach over all the introduced representations. However, non-normalised representation proved to be ineffective due to the inhomogeneity in the average flux, with clusters agglom-

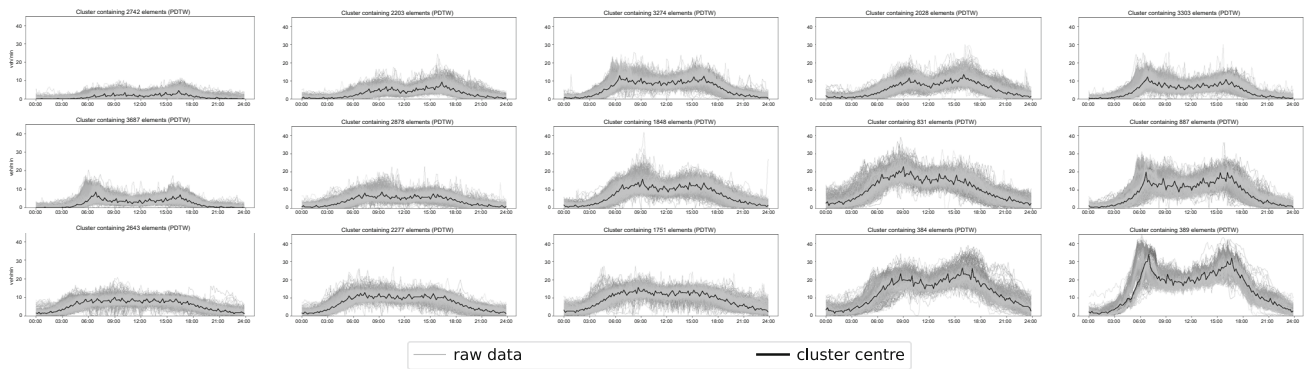


Fig. 8 Fuzzy c -means results of the PDTW representation when $c = 15$. Each panel represents a series (in grey) assigned to a cluster, with the corresponding centroid overlaid in black

erating low-flux series in a few clusters while ignoring most of the higher extremes.

For the remaining representations, we adopted the heuristic described in Sect. 3.1 to locate suitable dendrogram cuts (cf. Fig. 9). We also decided to filter clusters smaller than 1000 elements in order to exclude outliers from the clustering. Figure 10 presents the results obtained with SAX representation in terms of seven clusters. Differently from previous approaches, due to the applied normalisation and the outlier removal, we can clearly distinguish different shapes emerging while flux magnitude is almost completely invisible.

ESAX results are similar in shape and conclusions and are omitted for the sake of conciseness.

Multivariate analysis results

Considering classes and lanes individually allowed obtaining numerous different insights and results, unveiling patterns previously hidden by the natural aggregation considered. In what follows, we report some of the most interesting conclusions for each approach with no claim of completeness.

As an example, every clustering obtained with at least three clusters over type T1 (cf. Table 1) was able to almost perfectly sort out weekdays from weekends. When increasing the number of clusters, some combinations also divide Saturdays from Sundays, with holidays typically grouped with Sundays. This behaviour is intrinsic to the difference in traffic composition, with e.g. heavy vehicles nearly missing on non-working days. Figure 11 presents an example of clustering results achieved on T1 with 4 clusters showcasing this feature.

Approaches T2 and T3 both obtained similar results but with lower accuracy in distinguishing between weekdays and holidays; in particular, the aggregation of *light vehicles* on both lanes tended to create outlier clusters presenting extremely high light flux (mostly holidays, but without separation between Saturdays and Sundays).

4.2 Anomaly detection results

To build our anomaly detector, we selected from the previous section the three most promising clustering techniques, namely (i) the fuzzy clustering combined with PDTW (cf. Fig. 8), (ii) the hierarchical clustering with SAX (cf. Fig. 10), and (iii) the hierarchical clustering with ESAX. This provided us with $\ell = 3$ anomaly scores for a given series, each one providing a relevant contribution being tuned on different characteristics in the series. Particularly, partitioning on PDTW is more focused on the flux values (series height), HCA on SAX on the shape of the series, whereas HCA on ESAX on single-point anomalies (which go unnoticed by the not-Extended counterpart).

To combine the different anomaly scores, we implemented both the AGG and POS approaches introduced in Sect. 3.2. The results were empirically proved to be good at identifying the most anomalous series during a single day. Figure 12 provides a few representative sample results.

We also conducted an initial anomaly detection analysis using the results from multivariate clustering. To understand its potentialities, we ranked the multivariate datasets based on the reconstruction error. Since the original datasets were pruned of trivial anomalies, the focus shifted to the additional insights offered by this method on normal-looking series.

As expected, the multivariate approach highlighted different characteristics compared to the univariate approach, emphasising differences between classes. For instance, Fig. 13 shows a series that consistently ranks among the furthest from the corresponding centroid in the multivariate k -means analysis despite being pretty “normal” if considered in the classical univariate analysis. Here, in fact, the light vehicles are almost absent in the afternoon, while the heavy counterpart is almost absent in the morning, hence composing fluxes that both present a particular shape, which, surprisingly, balances out in the aggregated flux. A deeper analysis reveals the day to be a post-holiday.

Fig. 9 Plot of the number of p -significant clusters (clusters with more than p series assigned) obtained during the agglomeration procedure applied on SAX (left) and ESAX (right) representations. Dendrogram cuts detected by the heuristic are highlighted with a black line

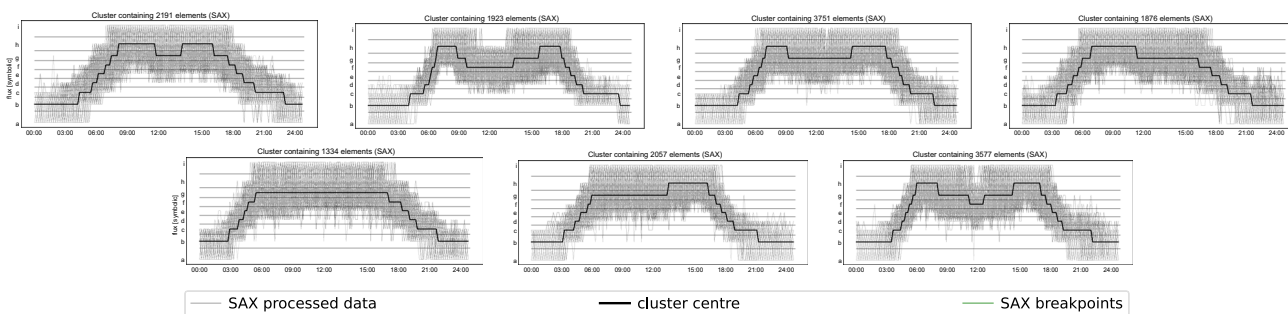
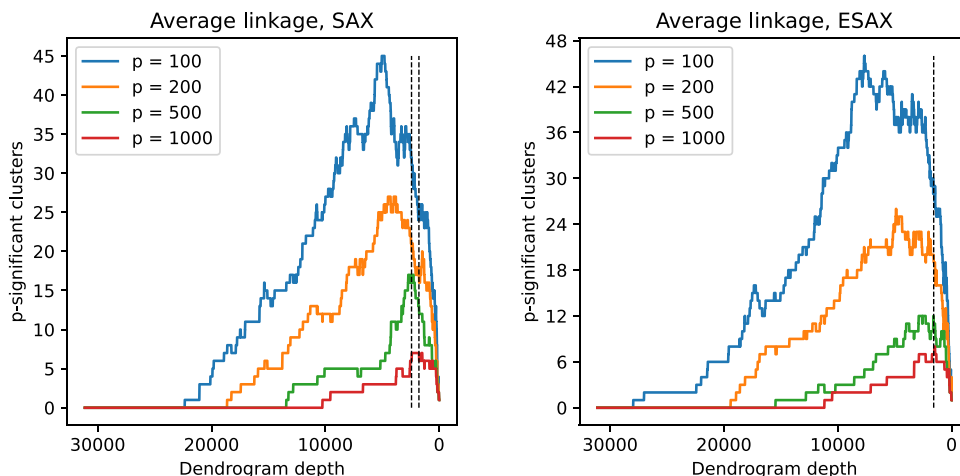


Fig. 10 Clustering results obtained cutting the dendrogram of SAX representation (cf. Fig. 9 left, right cut). Each panel represents series (in grey) assigned to a large cluster (> 1000 series), with the corresponding centroid overlaid in black (cf. Algorithm 1)

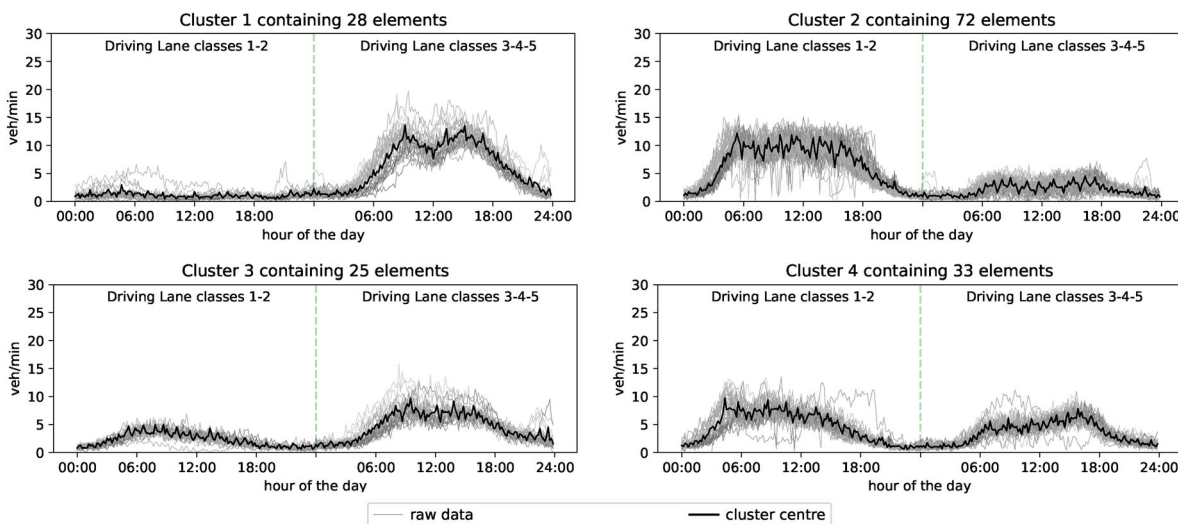


Fig. 11 Multivariate (T1) fuzzy c -means results of the PDTW representation when $c = 4$. Each panel represents series (in grey) assigned to a cluster, with the corresponding centroid displayed in black. Based

on the calendar days: Cluster 1 (top left) collects 1 (of 24) Saturday, all Sundays (23/23) and national holidays (4/4) contained in the dataset; Cluster 3 (bottom left) collects remaining Saturdays (23/24)

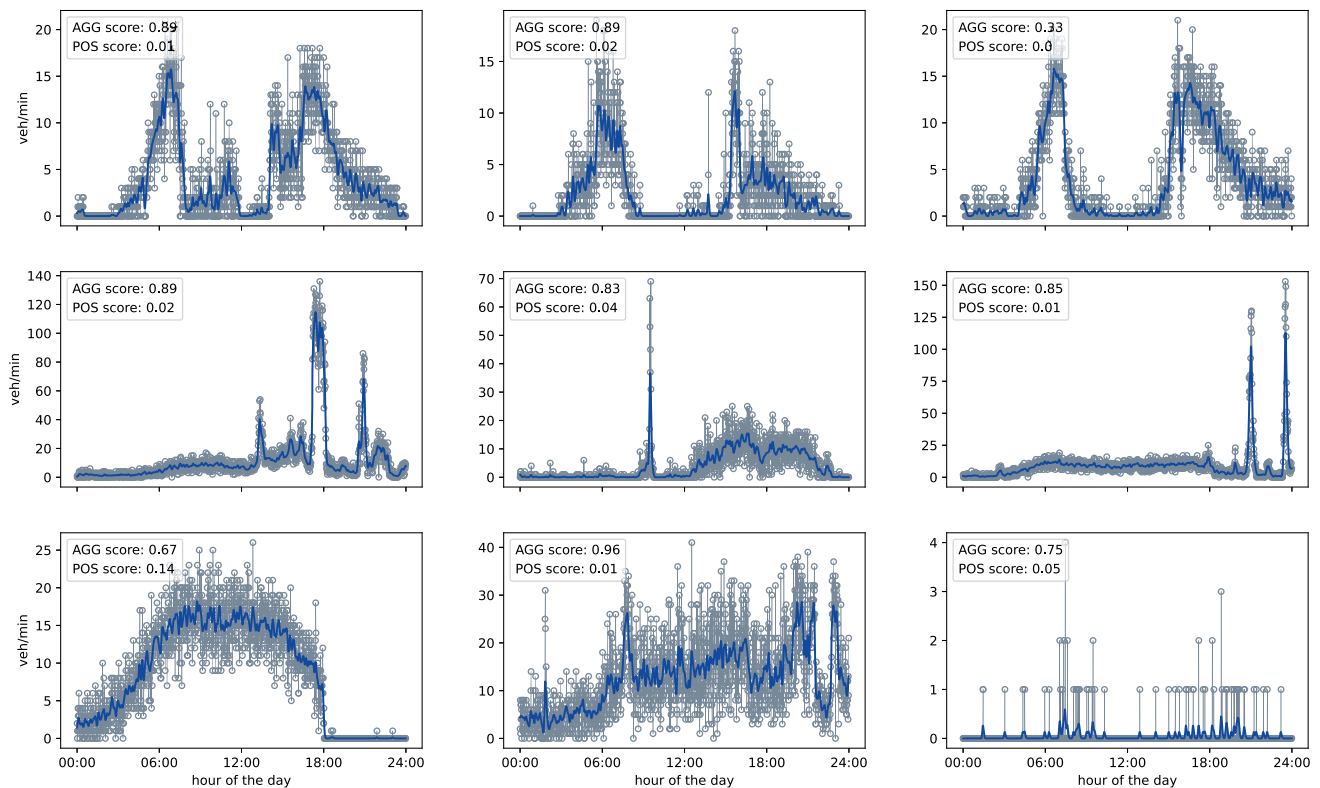


Fig. 12 Example of daily anomalies found using the proposed anomaly detection techniques. The AGG and POS scores are reported in the images. Do notice that, for the sake of readability, y-axes are not homogeneous. Also, scores are day dependent, so they are not directly comparable. The top row reports three probable traffic anomalies caused by long congestion events. The central row depicts three anomalies

caused by sensor failures communicating unfeasibly high flux spikes. The bottom row presents three different kinds of events: (left) a sensor abruptly stopping reporting vehicles, (centre) unstable traffic conditions with abnormal flux (in particular at night time), and (right) a very low traffic reading either ascribed to sensor malfunctioning or to lane closure

While such cases do exist, their presence in the dataset seems to be limited, with no particular applicative appeal. As a consequence, we decided not to proceed with further analysis.

Injecting noisy signals

Given that our time series data are inherently noisy—and that such noise may originate not only from environmental variability but also from faulty or misaligned sensors—we conducted a study to evaluate how white noise affects the daily anomalies detected by our methods.

To this end, we analysed a subset of the original dataset comprising all series recorded in December 2022. White noise was injected into series one at a time, by adding point-wise normally distributed values with mean 0 and variance σ^2 , $\sigma^2 \in [0, 10]$, and by taking the absolute integer value of the result. Daily anomalous series are then re-evaluated and compared with the original daily ranking. Results are shown in Fig. 14.

Overall, when considering originally non-anomalous series, a white noise with $\sigma^2 = 4.4 \pm 2.9$ is required (on average) for the series to become anomalous, a quite significant

modification when directly looking at the series. Conversely, when series $s \in \mathcal{A}_d$ are taken into account, three different behaviours might arise: (i) s is removed from \mathcal{A}_d in the repeated analysis, (ii) at least another anomaly is removed from \mathcal{A}_d , and (iii) at least a new anomaly appear. Unsurprisingly, the probability for the anomaly to be removed is smaller the higher the variance (maximum 4% with $\sigma^2 = 1$). Conversely, the higher the variance, the higher anomaly score, the higher the probability of other anomalies being removed due to the comparative nature of the score. Finally, in few occurrences (< 11%) the noise lowers the anomaly scores, actually allowing more anomalies to appear.

5 Conclusions and future work

In this paper, we introduced a comprehensive framework for analysing time series data using clustering techniques, detailing the essential preprocessing steps, similarity measures, clustering methodologies, and evaluation techniques. We draw our focus on the interaction between these com-

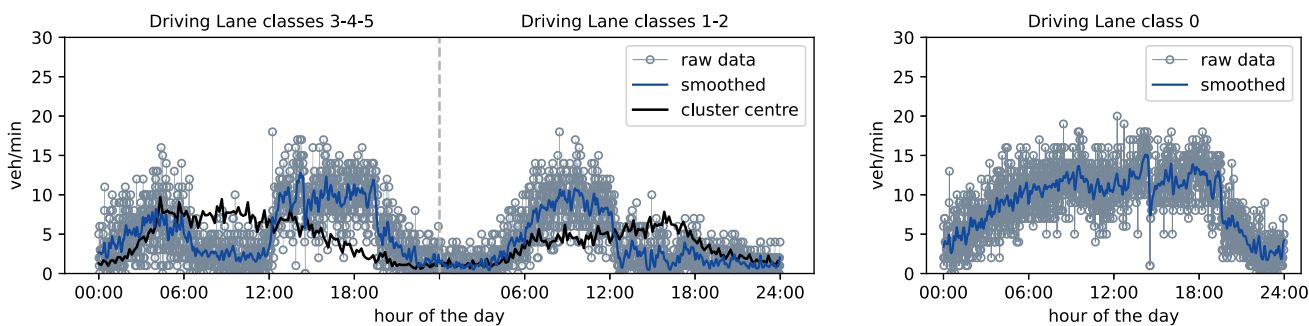


Fig. 13 Example of a traffic anomaly detectable with multivariate analysis only. Flux of light and heavy vehicles (left panel) surprisingly sum to a regular total flux (right panel). Anomaly is detected with fuzzy *c*-means over PDTW representation (cf. Fig. 11, Cluster 4)

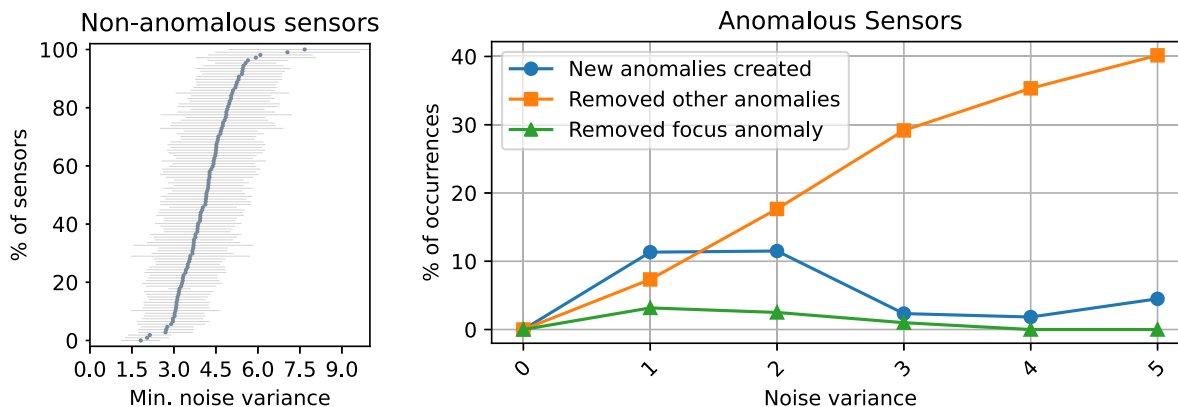


Fig. 14 Effect over daily detected anomalies when white noise is injected on a non-anomalous (left) or anomalous (right) series. On the left, series (organised by sensor, y-axis) are plotted against the minimum required variance (x-axis) for them to become anomalies (average

highlighted). On the right, probability (y-axis) that the noise injection removes the targeted anomaly, removes other anomalies, or adds novel anomalies w.r.t. the noise variance (x-axis)

ponents and the necessary adaptations required to obtain meaningful results, such as the symbolic averaging method described in Appendix A.

We applied our methodology to a real-world traffic dataset, demonstrating the strengths and limitations of various clustering techniques and how dataset characteristics influence outcomes. Additionally, we explored anomaly detection, both as a theoretical framework and through a practical implementation, leveraging clustering results for an efficient identification of irregular traffic patterns.

Our results showed that, for our dataset, applying fuzzy *c*-means clustering using PDTW is the best choice in terms of general separation of the dataset, when the absolute values of the series are taken into consideration. Conversely, by introducing a normalisation step, combining SAX/ESAX representations with HCA yields the best results when judged by the ability to discern shapes. While these final considerations are valid specifically for our dataset, we also gained more general conclusions, such as the incompatibility between symbolic representations and partitioning clustering.

Our findings offer valuable insights into creating a pipeline for time series clustering and anomaly detection, but they do represent only an initial step. As an example, although the developed anomaly detection technique proved to be very useful for practical purposes to automatically identify anomalies on a daily basis, some important limitations remain, especially at the academic level. One of the main issues is certainly the distinction between anomalies due to sensor malfunctions (which require maintenance intervention) and those due to particular traffic situations (which could require an intervention at the road level). Secondly, the proposed method does not sharply label the data as anomalous/non-anomalous, but rather lists them in order of the degree of anomaly, then considers the most anomalous for the day only. This leaves open the problem of finding a sweet alert threshold, also considering that a maximum number of critical situations can be reasonably addressed each day. Also, let us mention that certain anomalies are clearly due to human intervention (e.g., a programmed road closure) and should be automatically discarded. Finally, we did not implement any automatic online learning mechanism,

meaning that the model does not adapt to evolving data distributions or newly emerging patterns. This inherently limits the long-term robustness of the approach, as static models may become less effective when confronted with gradual behavioural shifts, seasonal changes, or structural modifications in the monitored system.

Future research should expand on the multivariate approach and refine anomaly detection, focusing on differentiating between sensor malfunctions and genuine traffic anomalies. Additionally, a more comprehensive analysis on anomaly detection scoring frameworks would enhance results credibility, implicitly requiring the standardisation and labelling of the dataset itself. Further investigation is also required to establish which clustering methods yield the most reliable results across different traffic scenarios and whether different metrics (warping and not warping) might lead to more reliable results. Finally, online learning approaches, like dynamic clustering, are also a field worth of interest.

A Averaging symbolic series

In this appendix, we introduce an approach which is, to the best of our knowledge, previously unpublished to evaluate the centroid of a given cluster of symbolic representations $C = \{\hat{s}_1, \dots, \hat{s}_m\}$ (with each series $\hat{s}_j = (x_1^{(j)}, \dots, x_w^{(j)})$) relatively to MINDIST (4). Algorithm 1 summarises the process.

Algorithm 1 Symbolic_Average(c)

Require: A set $\{\hat{s}_1, \dots, \hat{s}_m\}$ of symbolic w -length series \hat{s}_i of alphabet \mathcal{A}

Ensure: An average symbolic series μ .

```

1: for  $i = 1, 2, 3, \dots, w$  do
2:   for  $\alpha$  in  $\mathcal{A}$  do
3:      $n_\alpha = \#\{x_i^{(j)} \text{ for } j \in \{1, \dots, m\} \text{ if } x_i^{(j)} = \alpha\}$ 
4:   end for
5:    $\mu_i = \operatorname{argmin}_{\beta \in \mathcal{A}} \left( \sum_{\alpha \in \mathcal{A}} (n_\alpha \cdot (\operatorname{dist}(\alpha, \beta))^2) \right)$ 
6: end for

```

The fundamental part of the proposed algorithm lies in the analogy with the classical version in \mathbb{R}^w . In fact, the approximation of the centroid $\mu = \{\mu_1, \dots, \mu_w\}$, which is typically obtained by averaging the time series, can be reformulated as the search problem of finding $\mu \in \mathbb{R}^n$ as

$$\mu = \operatorname{argmin}_r \frac{1}{m} \sum_{i=1}^m L^2(s_i, r)^2 \quad (7)$$

Given a set of symbolic representations $\{\hat{s}_1, \dots, \hat{s}_m\}$ and the distance measure MINDIST, finding the sequence $r =$

$\{r_1, \dots, r_w\}$ such as

$$r = \operatorname{argmin}_{s \in \mathcal{S}} \frac{1}{m} \sum_{i=1}^m (\operatorname{MINDIST}(\hat{s}_i, s))^2 \quad (8)$$

where \mathcal{S} is the set of all possible symbolic sequences $s = \{x_1, \dots, x_w\}$ of length w , can be seen by MINDIST definition as

$$r = \operatorname{argmin}_{s \in \mathcal{S}} \sum_{i=1}^m \left(\sqrt{\frac{n}{w} \sum_{j=1}^w (\operatorname{dist}(\hat{x}_j^{(i)}, x_j))^2} \right)^2 \quad (9)$$

which equals

$$\begin{aligned} r &= \operatorname{argmin}_{s \in \mathcal{S}} \sum_{i=1}^m \left(\frac{n}{w} \sum_{j=1}^w (\operatorname{dist}(\hat{x}_j^{(i)}, x_j))^2 \right) \\ &= \operatorname{argmin}_{s \in \mathcal{S}} \sum_{i=1}^m \left(\sum_{j=1}^w (\operatorname{dist}(\hat{x}_j^{(i)}, x_j))^2 \right). \end{aligned} \quad (10)$$

Since every $\operatorname{dist}()$ computation is independent, we can rewrite (10) as

$$r = \operatorname{argmin}_{s \in \mathcal{S}} \sum_{j=1}^w \left(\sum_{i=1}^m (\operatorname{dist}(\hat{x}_j^{(i)}, x_j))^2 \right). \quad (11)$$

Now, every j computation is independent of the other, which means that every component of the first sum can be optimised separately. Hence, the j -th component of r equals the letter $\beta \in \mathcal{A}$ that minimises the internal sum, i.e.

$$r_j = \operatorname{argmin}_{\beta \in \mathcal{A}} \sum_{i=1}^m (\operatorname{dist}(\hat{x}_j^{(i)}, \beta))^2. \quad (12)$$

Since both $\hat{x}_j^{(i)}$ and β are in the same finite alphabet \mathcal{A} , the number of possible distances is finite; hence, we can rewrite (12) as

$$r_j = \operatorname{argmin}_{\beta \in \mathcal{A}} \sum_{\alpha \in \mathcal{A}} (n_\alpha \cdot (\operatorname{dist}(\alpha, \beta))^2), \quad (13)$$

where n_α is the cardinality of $\{\hat{x}_j^{(i)} \mid i = 1, \dots, m \mid \hat{x}_j^{(i)} = \alpha\}$, matching the thought process behind Algorithm 1. Using (13) instead of (12) reduces the number of distance computations, although increasing the total number of simple operations.

A weighted formulation (for the fuzzy implementation) is possible as well. Assuming that a weight $w_i \in \mathbb{R}$ is associated with every time series \hat{s}_i , we can redefine our objective

function (8) for the average r as

$$\begin{aligned} r &= \operatorname{argmin}_{s \in \mathcal{S}} \frac{\sum_{i=1}^m w_i \cdot (\text{MINDIST}(\hat{s}_i, s))^2}{\sum_{i=1}^m w_i} \\ &= \operatorname{argmin}_{s \in \mathcal{S}} \sum_{i=1}^m w_i \cdot (\text{MINDIST}(\hat{s}_i, s))^2. \end{aligned} \quad (14)$$

Following similar steps to (8)–(12), we obtain

$$r_j = \operatorname{argmin}_{\beta \in \mathcal{A}} \sum_{i=1}^m w_i \cdot (\text{dist}(\hat{x}_j^{(i)}, \beta))^2. \quad (15)$$

which can be rewritten as

$$r_j = \operatorname{argmin}_{\beta \in \mathcal{A}} \sum_{\alpha \in \mathcal{A}} (n_{\alpha, w} \cdot (\text{dist}(\alpha, \beta))^2), \quad (16)$$

where $n_{\alpha, w} = \text{sum}\{w_i \mid i = 1, \dots, m \mid \hat{x}_j^{(i)} = \alpha\}$.

Given that Algorithm 1 is based on the Euclidean average, one could suggest that averaging time series before changing the representation into SAX could return the same results. However, due to the nature of SAX, this is simply not true in a general case.

Acknowledgements The authors want to warmly thank Andrea Lisi for his support in analysing and understanding data, as well as improving the quality of the dataset. The authors also want to thank Paolo Ranut, who supported this research on behalf of the motorway company Autostrade Alto Adriatico S.p.A.

All authors are members of the Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica (GNCS-INdAM).

Author Contributions D.M. prepared the first draft, performed the literature review, developed the code, conducted initial testing, extracted the results, and created the graphics. E.O. initially analysed the dataset for clustering and anomaly detection, supervised the developments, provided the initial code, and contributed to the literature review, writing, and result analysis and discussion. E.C. supervised the overall research, conceptualised the study, secured funding, and managed the collaboration with the data-providing company. All authors validated the results with the assistance of the data-providing company. All authors reviewed and approved the final manuscript.

Fundings Open access funding provided by Università degli Studi Roma Tre within the CRUI-CARE Agreement. This work was partially funded by Autostrade Alto Adriatico E.C. would like to thank the Italian Ministry of University and Research (MUR) for supporting this research with funds coming from PRIN Project 2022 PNRR (No. 2022XJ9SX, entitled “Heterogeneity on the road - Modeling, analysis, control”). This study was carried out within the Spoke 7 of the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-Generation EU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 D.D. 1033 17/06/2022, CN00000023). This manuscript reflects only the authors’ views and opinions. Neither the European Union nor the European Commission can be considered responsible for them.

Data Availability Data and related codes are not publicly available.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Mitsa, T.: Temporal Data Mining. Chapman and Hall/CRC, New York (2010). <https://doi.org/10.1201/9781420089776>
- Briani, M., Cristiani, E., Onofri, E.: Inverting the fundamental diagram and forecasting boundary conditions: how machine learning can improve macroscopic models for traffic flow. *Adv. Comput. Math.* **50**(6), 115 (2024). <https://doi.org/10.1007/s10444-024-10206-8>
- Aghabozorgi, S., Seyed Shirkhorshidi, A., Ying Wah, T.: Time-series clustering a decade review. *Inf. Syst.* **53**, 16–38 (2015). <https://doi.org/10.1016/j.is.2015.04.007>
- Warren Liao, T.: Clustering of time series data—a survey. *Pattern Recognit.* **38**(11), 1857–1874 (2005). <https://doi.org/10.1016/j.patcog.2005.01.025>
- MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, pp. 281–297 (1967). <https://www.cs.cmu.edu/~bhiksha/courses/mlsp.fall2010/class14/macqueen.pdf>
- Bezdek, J.C.: Cluster validity with fuzzy sets. *J. Cybern.* **3**(3), 58–73 (1973). <https://doi.org/10.1080/01969727308546047>
- Johnson, S.C.: Hierarchical clustering schemes. *Psychometrika* **32**(3), 241–254 (1967). <https://doi.org/10.1007/bf02289588>
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD’96, pp. 226–231. AAAI Press, Washington, DC, 20004, USA (1996). <https://dl.acm.org/doi/10.5555/3001460.3001507>
- Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006). <https://link.springer.com/9780387310732>
- Aggarwal, C.C., Yu, P.S., Han, J., Wang, J.: A Framework for Clustering Evolving Data Streams, pp. 81–92. Elsevier, San Francisco (2003). <https://doi.org/10.1016/B978-012722442-8/50016-1>
- Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **26**(1), 43–49 (1978). <https://doi.org/10.1109/tassp.1978.1163055>

12. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 359–370. AAAI Press, Washington, DC 20004, USA (1994). <https://dl.acm.org/doi/10.5555/3000850.3000887>
13. Keogh, E., Lin, J., Fu, A.: HOT SAX: efficiently finding the most unusual time series subsequence. In: Fifth IEEE International Conference on Data Mining (ICDM'05), pp. 226–233. IEEE, Los Alamitos, CA 90720-1314 (2005). <https://doi.org/10.1109/icdm.2005.79>
14. Fulcher, B.D., Jones, N.S.: Highly comparative feature-based time-series classification. *IEEE Trans. Knowl. Data Eng.* **26**(12), 3026–3037 (2014). <https://doi.org/10.1109/tkde.2014.2316504>
15. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD09, pp. 947–956. ACM, New York (2009). <https://doi.org/10.1145/1557019.1557122>
16. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Inf. Syst.* **3**(3), 263–286 (2001). <https://doi.org/10.1007/pl00011669>
17. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD00, pp. 285–289. ACM, New York (2000). <https://doi.org/10.1145/347090.347153>
18. Chu, S., Keogh, E., Hart, D., Pazzani, M.: Iterative deepening dynamic time warping for time series. In: Proceedings of the 2002 SIAM International Conference on Data Mining, pp. 195–212. Society for Industrial and Applied Mathematics, Philadelphia (2002). <https://doi.org/10.1137/1.9781611972726.12>
19. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. DMKD03, pp. 2–11. ACM, New York (2003). <https://doi.org/10.1145/882082.882086>
20. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. *Data Min. Knowl. Disc.* **15**(2), 107–144 (2007). <https://doi.org/10.1007/s10618-007-0064-z>
21. Lkhagva, B., Suzuki, Y., Kawagoe, K.: Extended SAX: extension of symbolic aggregate approximation for financial time series data representation. In: Proceeding of IEICE the 17th Data Engineering Workshop (2006). <https://www.ieice.org/~de/DEWS/DEWS2006/doc/4A-i8.pdf>
22. Shieh, J., Keogh, E.: iSAX: indexing and mining terabyte sized time series. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 623–631 (2008). <https://doi.org/10.1145/1401890.1401966>
23. Sun, Y., Li, J., Liu, J., Sun, B., Chow, C.: An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing* **138**, 189–198 (2014). <https://doi.org/10.1016/j.neucom.2014.01.045>
24. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987). [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
25. Staples, L., Ring, J., Fontana, S., Stradwick, C., DeMaio, J., Ray, H., Zhang, Y., Zhang, X.: Reproducible clustering with non-Euclidean distances: a simulation and case study. *Int. J. Data Sci. Anal.* (2023). <https://doi.org/10.1007/s41060-023-00429-1>
26. Wu, K.-L., Yang, M.-S.: A cluster validity index for fuzzy clustering. *Pattern Recognit. Lett.* **26**(9), 1275–1291 (2005). <https://doi.org/10.1016/j.patrec.2004.11.022>
27. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. *Pattern Recognit.* **46**(1), 243–256 (2013). <https://doi.org/10.1016/j.patcog.2012.07.021>
28. Onofri, E.: ML- and graph-based methods for data analysis and prediction: applications in vehicular traffic, pedestrian dynamics, and computational biology. Ph.D. Thesis, University Roma Tre (2023). <https://doi.org/10.13140/RG.2.2.32948.36489>
29. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.* **54**(3), 1–33 (2021). <https://doi.org/10.1145/3444690>
30. Izakian, H., Pedrycz, W.: Anomaly detection in time series data using a fuzzy c-means clustering. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 1513–1518 (2013). <https://doi.org/10.1109/IFSA-NAFIPS.2013.6608627>
31. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. *ACM SIGMOD Rec.* **29**(2), 93–104 (2000). <https://doi.org/10.1145/335191.335388>
32. Chen, Z., Yeo, C.K., Lee, B.S., Lau, C.T.: Autoencoder-based network anomaly detection. In: 2018 Wireless Telecommunications Symposium (WTS), pp. 1–5. IEEE, Los Alamitos, CA 90720-1314 (2018). <https://doi.org/10.1109/wts.2018.8363930>
33. Shyu, M.-L., Chen, S.-C., Sarinapakorn, K., Chang, L.: Principal Component-based Anomaly Detection Scheme. Chap. 3–3, pp. 311–329. Springer, Heidelberg (2005). https://doi.org/10.1007/11539827_18
34. Diab, D.M., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K.G., Ghafir, I.: Anomaly detection using dynamic time warping. In: 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 193–198. IEEE, Los Alamitos, CA 90720-1314 (2019). <https://doi.org/10.1109/cse/euc.2019.00045>
35. Markovic, T., Dehlaghi-Ghadim, A., Leon, M., Balador, A., Punnekatt, S.: Time-series anomaly detection and classification with long short-term memory network on industrial manufacturing systems. In: Proceedings of the 18th Conference on Computer Science and Intelligence Systems. FedCSIS 2023, vol. 35, pp. 171–181. IEEE, Los Alamitos, CA 90720-1314 (2023). <https://doi.org/10.15439/2023f5263>
36. Ganapathy, J., García Márquez, F.P.: Data mining and information technology in transportation—a review. In: Proceedings of the Fifteenth International Conference on Management Science and Engineering Management, pp. 849–855. Springer, Cham, Switzerland (2021). https://doi.org/10.1007/978-3-030-79206-0_64
37. Banihosseini, M., Baradaran, V., Doroudyan, M.H.: Clustering approach to identifying and analyzing the traffic conditions: a novel hybrid cloud density and fuzzy clustering algorithm. *Int. J. Intell. Transp. Syst. Res.* (2024). <https://doi.org/10.1007/s13177-024-00417-0>
38. Reyes, G., Tolozano-Benites, R., Lanzarini, L., Estrebow, C., Barviera, A.F., Barzola-Monteses, J.: Methodology for the identification of vehicle congestion based on dynamic clustering. *Sustainability* **15**(24), 16575 (2023). <https://doi.org/10.3390/su152416575>
39. Reyes, G., Tolozano-Benites, R., Lanzarini, L., Estrebow, C., Barviera, A.F., Barzola-Monteses, J.: Method for the identification and classification of zones with vehicular congestion. *ISPRS Int. J. Geo Inf.* **13**(3), 73 (2024). <https://doi.org/10.3390/ijgi13030073>
40. Medina-Salgado, B., Sánchez-DelaCruz, E., Pozos-Parra, P., Sierra, J.E.: Urban traffic flow prediction techniques: a review. *Sustain. Comput. Inform. Syst.* **35**, 100739 (2022). <https://doi.org/10.1016/j.suscom.2022.100739>
41. Weijermars, W., Berkum, E.: Analyzing highway flow patterns using cluster analysis. In: Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005, pp. 831–836. IEEE, Los Alamitos, CA 90720-1314 (2005). <https://doi.org/10.1109/itsc.2005.1520157>

42. Asadi, R., Regan, A.: Spatio-temporal clustering of traffic data with deep embedded clustering. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility. SIGSPATIAL'19, pp. 45–52. ACM, New York (2019). <https://doi.org/10.1145/3356995.3364537>
43. Toshniwal, D., Chaturvedi, N., Parida, M., Garg, A., Choudhary, C., Choudhary, Y.: Application of clustering algorithms for spatio-temporal analysis of urban traffic data. *Transp. Res. Procedia* **48**, 1046–1059 (2020). <https://doi.org/10.1016/j.trpro.2020.08.132>
44. Li, M., Zhu, Y., Zhao, T., Angelova, M.: Weighted dynamic time warping for traffic flow clustering. *Neurocomputing* **472**, 266–279 (2022). <https://doi.org/10.1016/j.neucom.2020.12.138>
45. Keogh, E., Ratanamahatana, A.: Everything you know about dynamic time warping is wrong. In: 3rd Workshop on Mining Temporal and Sequential Data, in Conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA, vol. 1, pp. 1–11 (2004). <https://api.semanticscholar.org/CorpusID:2599643>
46. Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., Yurchak, R., Rußwurm, M., Kolar, K., Woods, E.: Tslern, a machine learning toolkit for time series data. *J. Mach. Learn. Res.* **21**(118), 1–6 (2020). <https://jmlr.org/papers/v21/20-091.html>
47. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognit.* **44**(3), 678–693 (2011). <https://doi.org/10.1016/j.patcog.2010.09.013>
48. Arthur, D., Vassilvitskii, S.: k-means++: the of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007). <https://dl.acm.org/doi/10.5555/1283383.1283494>
49. Winkler, R., Klawonn, F., Kruse, R.: Fuzzy c-means in high dimensional spaces. *Int. J. Fuzzy Syst. Appl.* **1**(1), 1–16 (2011). <https://doi.org/10.4018/ijfsa.2011010101>
50. Stetco, A., Zeng, X.-J., Keane, J.: Fuzzy C-means++: Fuzzy C-means with effective seeding initialization. *Expert Syst. Appl.* **42**(21), 7541–7548 (2015). <https://doi.org/10.1016/j.eswa.2015.05.014>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.