University of Padova

Department of Mathematics

Master Thesis in Data Science

User Profiling at a Music Festival from Attendees' Mobility Data

Supervisor
Michele Rossi
University of Padova

Co-supervisor
Patricio Reyes
Barcelona Supercomputing Center

Master Candidate
Camilo Betancourt Nieto

Academic Year
2024-2025

To the people I love: my parents, my brother and Estefanía.

Abstract

Clustering techniques are used in different domains to uncover patterns that would otherwise remain hidden in complex, unstructured datasets, supporting practical applications such as customer profiling, social behavior analysis, and traffic flow optimization. However, these methods are not typically tailored to the study of human mobility in events held at designated locations, such as music festivals and conventions, where the combination of activities and spatio-temporal dynamics introduces additional layers of complexity to the analysis. This study presents customized preprocessing and trajectory clustering adaptations for interpreting anonymous Wi-Fi traces of event attendees, addressing context-specific challenges such as unidentified sources of signals, uneven sample rates, and noisy sequences, all within an unsupervised setting with no ground truth. This strategy is compared with a network science approach that applies community detection to bipartite graphs built from implicit attendee feedback, discussing the distinct perspectives offered by each method. The clustering methods identified groups of attendees who stayed primarily within the two main audience zones and others with more exploratory behavior. Among the exploratory participants, some followed consistent movement patterns between venue areas, while others exhibited more irregular trajectories. These clusters, which also reflect musical preferences, provided different insights from those of the community detection techniques, which identified smaller groups with a stronger focus on music preference. These findings contribute to the understanding of participant behavior and present a methodological approach adaptable to similar event settings.

Contents

List of figures

List of tables

List of acronyms

GMM	Gaussian Mixture Models
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
HDBSCAN	Hierarchical DBSCAN
DTW	Dynamic Time Warping
LCSS	Longest Common Subsequence
DBA	DTW Barycenter Averaging
MIR	Music Information Retrieval
TID	Trajectory Identifier
ECDF	Empirical Cumulative Distribution Function
PCA	Principal Component Analysis
CTWE	Contemporary Time Window Euclidean
DBCV	Density-Based Clustering Validation Index
UMAP	Uniform Manifold Approximation and Projection
MDS	Multidimensional Scaling
t-SNE	t-distributed Stochastic Neighbor Embedding
BCS	Balanced Clustering Score
ARI	Adjusted Rand Index
NMI	Normalized Mutual Information
AMI	Adjusted Mutual Information

1
Introduction

The wide variety of data sources related to moving objects and people, such as GPS trajectories, mobile phone logs, and geotagged social media posts, along with advances in data management and analytics techniques, has made spatio-temporal data mining a relevant area of research, enabling pattern discovery in applications such as urban planning, intelligent transportation systems, and marketing [1, 2, 3]. In this context, trajectory clustering constitutes a useful tool for detecting common mobility trends and grouping spatio-temporal instances that would otherwise be difficult to analyze and represent.

However, trajectory clustering techniques are typically studied in open-world settings, such as urban environments, where movement is relatively unrestricted and spans broad spatial and temporal scales. These methods are not explicitly adapted for settings like music festivals, which take place in more localized environments with a limited set of locations, and where movement patterns are influenced by factors such as the event's lineup and venue layout.

On the other hand, while previous research has analyzed Wi-Fi traces at a music festival to capture musical similarity and measure attendees' preferences based on their physical attendance [4], it does not center on other spatio-temporal nuances that trajectory data could reveal.

This work explores the use of trajectory clustering techniques in the context of a music festival to evaluate their effectiveness in uncovering meaningful patterns in attendee behavior. While building on classical trajectory clustering techniques, this study introduces adaptations in both preprocessing steps and modeling frameworks to better accommodate the specific characteristics of the festival setting. In particular, it evaluates the ability of this approach to mine noisy Wi-Fi traces with sparse trajectories and no explicit feedback (ground truth) while identifying relevant movement patterns and distinguishing characteristic attendee profiles.

Additionally, this strategy was compared to a network science approach with community detection on bipartite graphs built from attendee-event interactions, building on prior studies of implicit feedback while offering a distinct analytical perspective. This comparison enabled an analysis of the similarities between both techniques and a discussion of the distinct nuances each method reveals.

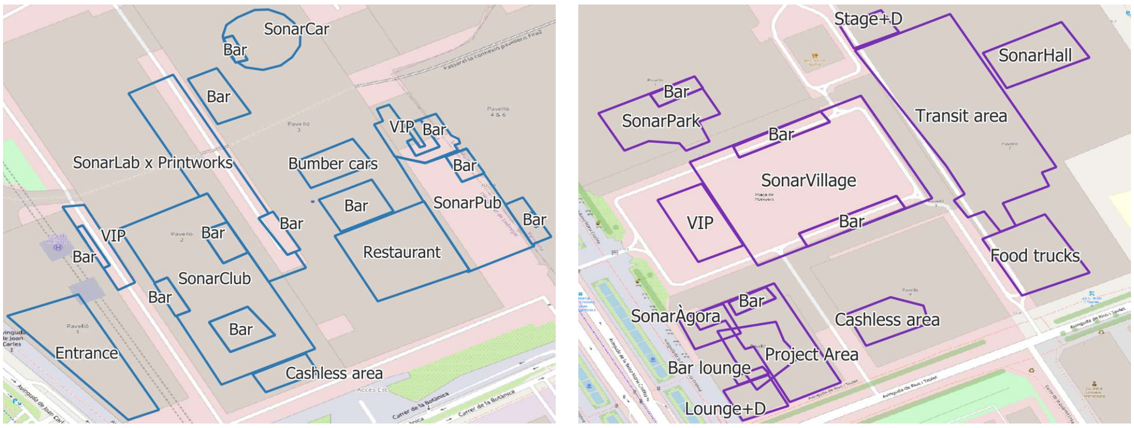
This thesis emphasizes interpretability and control over the design of modeling frameworks, aiming to identify the factors that shape the resulting clusters and understand the reasoning behind the results. Instead of relying on more abstract methods like neural architectures, which often perform well in large-scale data but can limit interpretability, this study applies handcrafted techniques suited for a moderate-scale dataset. As an investigation of a relatively unexplored and specific scenario, this work successfully uncovers meaningful patterns in attendees' behaviors, offering insights into the festival's audience and informing better event strategy and decision-making. It also helps to establish a foundation for future research in similar settings, opening opportunities to extrapolate its principles to more modern frameworks while offering insights into which modeling aspects are effective.

This thesis is structured as follows. Sec. 1.1 introduces the general context of the Sónar Festival (the event analyzed). Chapter 2 covers the theoretical background and related work, followed by Chapter 3, which details the methodology, including a detailed description of the preprocessing steps given their significance in this study. Chapter 4 presents and discusses the results, and Chapter 5 concludes with final remarks.

1.1 Sónar Festival Context

Sónar is a festival, held annually in Barcelona since 1994, that serves as a platform for electronic music and digital culture, focusing on emerging trends in music, performance, and technology.

The 2024 edition took place from June 13 to 15 and featured two distinct modalities: Sónar by Day and Sónar by Night, each held at a different venue. Sónar by Day was hosted at Fira Montjuïc from Thursday to Saturday, offering music performances, talks, and other activities from about 10:00 to midnight. In contrast, Sónar by Night was held at Fira Gran Vía on Friday and Saturday nights, exclusively showcasing music performances from 20:50 to 07:00 in the morning. The specific lineup of the 2024 edition can be consulted in the archived official schedule [5].



(a) Designated areas for Sónar by Night.

(b) Designated areas for Sónar by Day.

Figure 1.1: Layouts of Sónar by Day and Sónar by Night.

For each Sónar modality, specific locations within the venues were allocated for the festival, including audience zones corresponding to each stage, as well as other areas such as restaurant spaces, entrances, exhibition zones, and more. The layouts of Sónar by Night and Sónar by Day are shown in Fig.1.1.

This thesis is based on the Wi-Fi traces collected at the venues hosting Sónar 2024, which captured device-network interactions. While data was available and preprocessed for both Sónar modalities, the full analysis was conducted exclusively for Sónar by Night.

2

Background and Related Work

A brief exposition of key principles and works relevant to this study is presented in this chapter.

Foundational concepts related to trajectory clustering are summarized in Sections 2.1 and 2.2. Then, Section 2.3 outlines more specific approaches for this task, while Section 2.4 presents key ideas relevant to an alternative method focused on community detection in graphs. Finally, Section 2.5 describes an earlier study on mobility dynamics and musical preference inference, using data from a prior edition of the Sónar festival.

2.1 Overview of Relevant Clustering Methods

In general, clustering refers to the process of grouping a set of objects into clusters, such that objects within the same cluster are more similar to one another according to specified criteria, while objects in different clusters are more dissimilar. There are several approaches to this problem, so this section provides a brief overview of two families of clustering methods that are pertinent to my work.

2.1.1 Prototype-Based Clustering

Some clustering techniques aim to group data points around representative prototypes. These prototypes serve as the central elements of clusters, capturing the essence of the points assigned to them. Two common approaches for this kind of clustering are K-means and K-medoids, both of which partition the dataset into K clusters by minimizing a distance-based objective function.

The classical K-means algorithm starts by choosing some initial value for the set of $\boldsymbol{\mu}_k$ prototypes (centroids), and can be then split into two iterative phases: assigning each data point $\boldsymbol{x}_i$ to their closest centroid, and recomputing the prototypes as the mean of all points within each cluster [6]. What this procedure is doing is minimizing a distortion measure:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2, \tag{2.1}$$

where $r_{ik} = 1$ when $\boldsymbol{x}_i$ belongs to cluster k and equals zero otherwise.

This formulation uses the squared Euclidean distance as a measure of dissimilarity, which can make the clusters sensitive to outliers and may be limiting if the data cannot be easily represented as single points[6].

An alternative to address this issue is K-medoids, which allows the inclusion of a more general dissimilarity metric $\mathcal{V}$ in the distortion measure:

$$\tilde{J} = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \mathcal{V}(\boldsymbol{x}_i, \boldsymbol{\mu}_k). \tag{2.2}$$

In this case, the step that assigns each object to its corresponding cluster is analogous to the process in K-means. However, the phase that determines the prototype for each cluster can be more complex, since computing the mean might not be feasible. For this reason, it is common to choose one of the data points within the cluster to be $\boldsymbol{\mu}_k$. This involves searching over the N_k points in each cluster, which requires $O(N_k^2)$ evaluations of $\mathcal{V}$, but provides the algorithm with the flexibility to work with any dissimilarity metric, as long as it can be evaluated [6].

As a quick note, in addition to the strategies described above, there are other prototype-based methods that maximize the expected log-likelihood of a model that assigns data points to clusters with some probability [7]. These approaches, like the one involving Gaussian Mixture Models (GMM), allow for more flexible assignments, where data points can belong to multiple clusters with varying degrees of probability, making them useful for handling overlapping or ambiguous cluster boundaries.

It is noteworthy that the strategies discussed in this section require the number of clusters, K, to be predetermined. While techniques like the elbow method [8] help address this issue, it still imposes a challenge, especially when there is little prior knowledge about the data [9].

2.1.2 Density-Based Clustering

The methods presented in Section 2.1.1 may not be ideal for clusters of varying shapes and densities [7]. To address this, density-based algorithms offer an alternative, as they group data points based on their density within a given space.

An example of this type of approach is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [10]. This method considers that each cluster has a typical density of points significantly higher than the density outside the cluster, so it identifies regions of high point density as clusters and treats regions of low point density as noise. More specifically, the basic principle behind DBSCAN is that the neighborhood of points in a cluster, determined by a radius Eps, must contain at least a minimum number of points, $MinPts$.

This approach works with any distance function and, as shown in Fig. 2.1, allows for the identification of clusters with varying shapes, beyond the circular configurations typically produced by methods like K-means. Additionally, in contrast to the methods discussed in Section 2.1.1, this strategy does not require specifying the number of clusters, K, in advance.

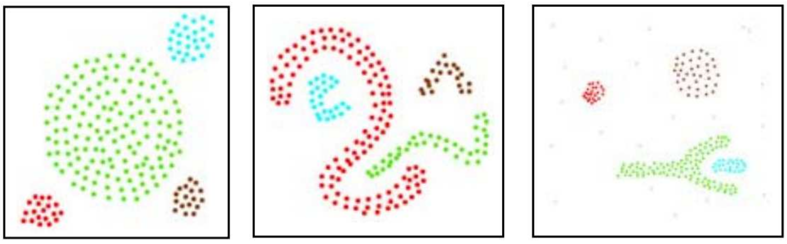


Figure 2.1: Example of clusters discovered by DBSCAN. Sourced from [10].

Nevertheless, since DBSCAN requires the parameters Eps and $MinPts$, the quality of the resulting clusters is highly sensitive to the choice of these values [9]. Hierarchical DBSCAN (HDBSCAN) is an alternative to alleviate this issue [11]. HDBSCAN can be conceptually seen as an exploration over all possible Eps values to discover clusters that persist for many values of this parameter. It builds a hierarchical cluster configuration across multiple distance thresholds and then refines it to select the most stable groupings.

By focusing on groups that remain consistent across different scales, this approach eliminates the need to explicitly choose Eps and effectively detects patterns with variable densities, a challenge for traditional DBSCAN [9, 12]. In contrast, it requires specifying a new, more intuitive parameter, $MinClusterSize$. Moreover, although the original HDBSCAN algorithm has a greater complexity than DBSCAN, there is an accelerated version [9] that performs comparably in metric spaces. For these reasons, HDBSCAN is a suitable choice for density-based clustering. Fig. 2.2 depicts a qualitative comparison between the clusters found by K-means, DBSCAN and HDBSCAN.

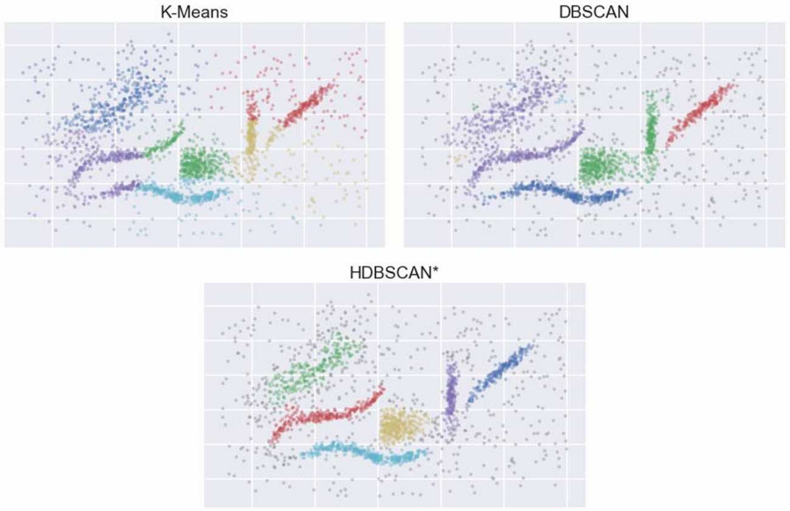


Figure 2.2: Qualitative comparison between different clustering techniques. Sourced from [9].

2.2 Human Mobility and Trajectory Data

This thesis utilizes a type of mobility data that captures the movements of individuals over a specific observation period. To provide clarity, it is helpful to mention some important definitions in this context [13, 3]. First, for an individual u, a trajectory S^u can be defined as a time-ordered sequence of the n_u visited spatio-temporal points: $S^u = (s_1, \dots, s_{n_u})$. Each point spatio-temporal point s_i is a tuple composed by spatial coordinates (e.g. latitude and longitude) and a timestamp.

It is worth mentioning that when trajectories are represented with geographical coordinates, the physical distance between two points s_a and s_b is often computed using the *haversine* formula for spherical surfaces [13]:

$$D_H = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta Lat}{2}\right) + \cos(Lat_{s_a}) \cdot \cos(Lat_{s_b}) \cdot \sin^2\left(\frac{\Delta Long}{2}\right)}\right), \tag{2.3}$$

where Lat and $Long$ represent the latitude and longitude of a point, $\Delta Lat = Lat_{s_b} - Lat_{s_a}$ and $\Delta Long = Long_{s_b} - Long_{s_a}$ are the differences in latitude and longitude between the two points, and r is the Earth's radius.

However, for small areas, the Earth's curvature is negligible, allowing for local plane approximations [14]. This makes calculations relative to a locally flat region appropriate for certain applications (e.g. the Euclidean distance).

Another key aspect of this type of mobility data is that not all recorded points provide meaningful information for analyzing a trajectory. In light of this, the concept of stay points becomes relevant [3]. Stay points are locations where an individual remains within a defined spatial threshold for a duration longer than a specified temporal threshold. These locations represent meaningful pauses in movement where an activity might take place. A stay point, $\dot{s}$, can be obtained as the geometric center of the trajectory points that satisfy both the spatial and temporal conditions.

There are numerous approaches to modeling mobility data, supporting tasks such as crowd flow prediction, individual next-location prediction, and generative tasks like flow or trajectory generation [13, 2]. However, this study specifically addresses *trajectory clustering*, which aims to identify groups of similar trajectories [15]. This method uncovers relevant patterns, deepens the understanding of the data, and potentially facilitates other applications. Moreover, the analysis seeks to contextualize mobility behaviors instead of solely examining raw movement traces, which constitutes a challenging task that demands a thoughtful choice of methodology.

2.3 Distance-Based Methods for Clustering Trajectory Data

Various existing strategies for clustering trajectories [16] rely on computing a distance or similarity metric, followed by the application of a clustering method, such as those described in Section 2.1. However, due to the spatio-temporal nature of trajectory data, selecting an appropriate criterion to measure similarity/distance is not trivial and extends beyond direct calculations between two points. As trajectories can be interpreted as sequences, similarity measures for time-series data, such as Dynamic Time Warping (DTW), Longest Common Subsequence (LCSS), and the method here referred to as *average contemporary distance*, have been used. A summary of their key aspects within the context of this study is provided below.

For clarity, the following descriptions refer to points in a two-dimensional space, but the basic principles can be extended to more dimensions.

2.3.1 Dynamic Time Warping

As a starting point, consider two discrete sequences, $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_m)$, and let δ be a distance function (e.g., the standard Euclidean distance). The distance between the two sequences can be thought of as an aggregation of the pairwise distances between corresponding elements [17]:

$$D(A, B) = \sqrt{\delta(a_1, b_1)^2 + \ldots + \delta(a_n, b_m)^2}. \tag{2.4}$$

However, this measure is quite rigid, as it requires sequences to have the same length (n and m should be equal) and only produces small distances for arrays that are perfectly aligned. It fails to capture similar patterns that are offset. To address these limitations, Dynamic Time Warping (DTW) was introduced for speech recognition applications [18]. This algorithm identifies the optimal alignment between sequences and recursively computes a cost based on both the intrinsic distance between individual elements and the cost of aligning them:

$$D(A_i, B_j) = \delta(a_i, b_j) + \min \left\{ \begin{matrix} D(A_{i-1}, B_{j-1}) \\ D(A_i, B_{j-1}) \\ D(A_{i-1}, B_j) \end{matrix} \right\}, \tag{2.5}$$

where A_i is the subsequence $(a_1, \ldots, a_i)$. The overall cost is obtained as $D(A_n, B_m)$. This approach allows for flexible quantification of dissimilarity between two sequences of different lengths and outputs a real number. Nevertheless, although it can be efficiently implemented using dynamic programming (and there are faster approximate variations, such as FastDTW [19]), it has $O(nm)$ time complexity and is generally more sensitive to noise than the LCSS distance [20], described in Section 2.3.2. An example of DTW alignment for two sequences is illustrated in Fig. 2.3.

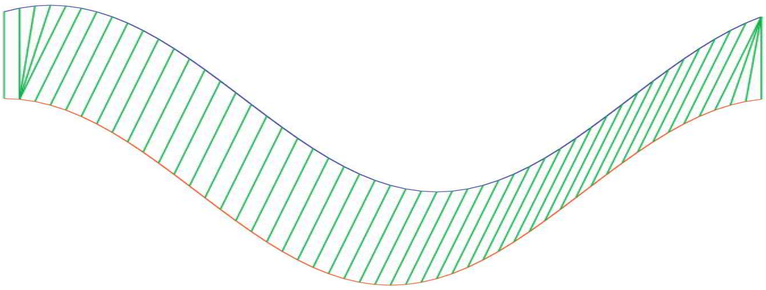


Figure 2.3: DTW alignment for two 1D sequences. Sourced from [17].

DTW provides a tool to uncover similarities in the trajectories' *patterns*, treating them as sequences and focusing on how similar their overall paths are rather than how perfectly synchronized they are. The measure obtained with this method can be used with clustering techniques that support custom dissimilarity metrics, such as K-medoids and density-based strategies. Moreover, a technique called DTW Barycenter Averaging (DBA) [17] allows for the computation of an average time series under the DTW distance measure, making it compatible with the K-means framework. In short, cluster centers are initialized using random or representative sequences. The average is then computed by aligning the time series within each cluster using DTW and updating the centroid with a barycenter average iteratively.

2.3.2 Longest Common Subsequence

In a similar direction to that of DTW, LCSS is another method to quantify similarity for time series, allowing sequences of different lengths to be compared [21]. In this case, given the two sequences A and B, the distance function δ, an integer γ and real number $0 < \epsilon < 1$, the $LCSS(A, B)$ can be expressed, in recursive form, as

$$\begin{cases} 0, & \text{if } A \text{ or } B \text{ is empty,} \\ 1 + LCSS_{\gamma,\epsilon}(\text{Head}(A), \text{Head}(B)), & \text{if } \delta(a_n - b_m) < \epsilon \text{ and } |n - m| \leq \gamma, \\ \max\big(LCSS_{\gamma,\epsilon}(\text{Head}(A), B), \\ \qquad LCSS_{\gamma,\epsilon}(A, \text{Head}(B))\big), & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\text{Head}(A) = \big((a_1, \ldots, a_{n-1})\big)$. This output can be then used to compute the similarity function

$$S_{LCSS}(\gamma, \epsilon, A, B) = \frac{LCSS_{\gamma,\epsilon}(A, B)}{\min(|A|, |B|)}, \tag{2.7}$$

which can be converted into a non-metric distance as $D_{LCSS} = 1 - S_{LCSS}(\gamma, \epsilon, A, B)$, ranging from 0 to 1. This formulation can also be implemented using dynamic programming with a time complexity of $O(nm)$.

Despite their similarities, LCSS presents key differences compared to DTW. First, it does not require matching every point in the sequences and allows for gaps in the alignment, making it more robust to noise in theory. Additionally, the LCSS distance involves two parameters, γ

and ϵ, which define the temporal and spatial thresholds for considering two points as a match. While they provide the user with control, they also introduce the need for careful tuning to ensure optimal performance, as their values influence the distance calculation.

On the other hand, there is not a well-established strategy analogous to DBA that combines LCSS with K-means, as updating centroids is not straightforward. However, LCSS can still be used with clustering algorithms like K-medoids and density-based approaches [20].

2.3.3 Average Contemporary Distance

Trajectories not only comprise the spatial dimensions of a moving object, and treating them solely as sequences of positions may overlook critical aspects of their dynamics. There are other approaches to measuring the distance between trajectories that aim to capture meaningful aspects of the time dimension in the analysis. In this context, Nanni and Pedreschi [22] proposed T-OPTICS, a specialized version of OPTICS (which is an adaptation of DBSCAN), tailored for trajectory data.

Specifically, in that work, the authors consider only *contemporary* observations of objects: they compare the objects' positions at a given moment and aggregate the set of distance values obtained in a time interval. This means that they discarded subsequence matching and alignment of trajectories, in contrast to what was earlier described. For them, the trajectory $\tau_u(\mathfrak{t})$ of an object u is a continuous function of time that, at time $\mathfrak{t}$, returns the position of the object (or individual). In this setting, the distance between trajectories is measured as the average distance δ between objects over the temporal interval $\mathfrak{T}$ when trajectories exist:

$$D(\tau_1, \tau_2)\big|_{\mathfrak{T}} = \frac{1}{|\mathfrak{T}|} \int_{\mathfrak{T}} \delta\big(\tau_1(\mathfrak{t}), \tau_2(\mathfrak{t})\big)\, d\mathfrak{t}. \tag{2.8}$$

This formulation requires that objects have a common temporal domain and is defined in continuous time, however, in practice, the distance can be obtained as a finite aggregation of distances between the available observations of trajectories τ_1 and τ_2.

The contemporary conditions imposed on the distance computation, could be advantageous for analyzing people's movement patterns during a music festival. In such contexts, the specific timing of events is crucial, and certain time segments may reveal key behavioral patterns based on attendees' mobility traces.

2.4 Community Detection on Graphs

Shifting the focus from the previous discussion, community detection on graphs presents an alternative approach for finding groups of related objects.

Graphs are commonly used to model relationships between instances, with nodes representing entities and edges representing their connections. A bipartite network is a type of graph where the set of nodes is divided into two distinct groups, and edges exist only between nodes of different groups. This structure is widely used to model relationships between two different types of entities, such as users and products, movies and actors, or words and documents [23].

What particularly concerns this study is how the interconnectivity structure of a network can reveal distinct communities based on the concept of modularity. Modularity quantifies the strength of a network's division into communities by comparing the density of connections within groups to those between groups. Given a partitioning of the network, a general definition of modularity is

$$Q = \frac{1}{2E} \sum_{ij} (M_{ij} - \xi P_{ij})\, I(c_i, c_j), \tag{2.9}$$

where M is the adjacency matrix, meaning M_{ij} represents the presence or weight of an edge between nodes i and j. The degree of a node is given by with $k_i = \sum_j M_{ij}$, while P_{ij} is the probability of connection under a null model, which represents the expected connectivity when rewiring the network randomly while preserving the degree distribution. Also, ξ is the resolution parameter, which can be used to influence the detected partition, leading to more or fewer communities. E is the total number of edges, while c_i and c_j are the communities of the nodes. The indicator function $I(c_i, c_j)$ equals 1 if i and j belong to the same community and 0 otherwise [23, 24]. Modularity has an upper bound of 1, with higher values indicating a stronger community structure. In practice, values around 0.3 to 0.7 suggest significant community structure, while larger values are rare [25].

For the special case of bipartite graphs, the restriction that connections cannot occur between nodes of the same type has led to different modularity definitions that account for this constraint when formulating the null model. A common example is Barber's modularity for bipartite graphs, given by [26, 23]:

$$Q_B = \frac{1}{F} \sum_{ij} \left(H_{ij} - \xi \frac{q_i d_j}{F} \right) I(c_i, c_j), \tag{2.10}$$

where H represents the bipartite adjacency matrix, while $q_i = \sum_j H_{ij}$, $d_j = \sum_i H_{ij}$, and $F = \sum_j d_j = \sum_i q_i$.

Modularity can be maximized to detect communities through the Louvain algorithm [27], a hierarchical method that greedily optimizes modularity in two iterative phases. In the first phase, each node is assigned to its own community, and nodes are dynamically reassigned to the neighboring community that yields the highest modularity gain. In the second phase, each detected community is collapsed into a single *super-node*, reducing the problem size. These phases repeat until modularity no longer improves. The algorithm typically runs in $O(n \log n)$. As a result, it detects distinct communities within the graph, revealing groups of nodes that are more strongly connected to each other than to the rest of the network.

2.5 Previous Study on Human Mobility and Musical Preference Using Sónar Festival Data

Although it takes a quite different approach from the one followed in this thesis, the work by Carrasco-Jiménez et al. [4] is an earlier study based on data from the 2015 edition of the Sónar festival. As in this case, they worked with anonymized Wi-Fi traces, but their focus was on assessing musical similarity between artists and analyzing the their performances in relation to the audience, using what they call *audience loyalty*.

First, the authors explain that in music information retrieval (MIR) applications, explicit user feedback, such as ratings, can be obtained in various ways, while implicit feedback is more challenging to capture. To address this, they propose a methodology to infer music preferences based on the mobility records of the festival attendees. This inferred measure of music preference is then applied to identify similar artists, which is a common task in the MIR field.

After matching the Wi-Fi traces with their associated stage and time during the festival, the authors computed a weighted user rating for each user-artist pair. This rating accounts for the proportion of time a user spent at an artist's concert relative to the total time the user spent at the festival.

From this, they constructed a graph with a node for each artist, and a link between nodes when artists had assistants in common. These links were weighted with a similarity measure *between artists*, derived from the set of users they had in common and the ratings inferred from those users' interactions with the respective artists. Next, the authors used community detection techniques to segment artists in three groups, based on the similarities revealed by the graph structure. The segmentation is shown in Fig. 2.4.

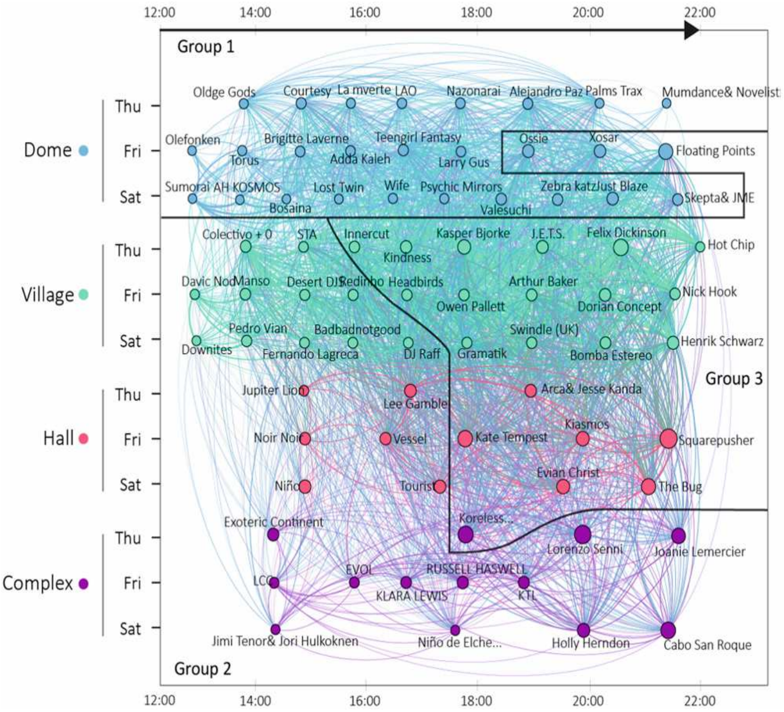


Figure 2.4: Artist groupings for Sónar 2015. The artists are separated by room (color), day (vertical axis), and performance time (horizontal), and linked by shared audience (width of lines). Sourced from [4].

Through a qualitative assessment of the groupings, the authors argue that group 1 consists of artists associated with attendees who preferred DJ-style electronic music, group 2 features the most experimental music of the festival, and group 3 corresponds to the most mainstream acts.

On a different note, the paper introduces a ranking system based on the weighted user rating they developed, which is compared to an alternative ranking that considers only raw audience size. This strategy brings attention to acts who might otherwise have gone unnoticed, those who successfully retain their audience (hence the term *audience loyalty*), even when performing on smaller stages.

Although the objectives and the methodologies of the described study differ significantly from the approach in this work (focusing on analyzing artist similarities and performances rather than clustering and analyzing *attendees'* mobility and activity behaviors), it is still informative to examine what has been done with a similar dataset.

3
Methodology

This chapter presents the processing pipeline developed for this analysis. It outlines the steps taken for data cleaning, specifies the modeling framework, describes the preparation of input features, and details the experimental setup.

3.1 Data Exploration and Cleaning

The exploratory and cleaning tasks were among the most challenging and time-intensive aspects of this project. The dataset contained significant noise, and isolating the core data for analysis required multiple stages, including some initial missteps. These steps were crucial for understanding the data and had a substantial impact on the project's outcomes. Given their importance and to ensure transparency, this section provides an intentionally detailed account of the process, including dataset description, identification of non-relevant devices, and noise removal from trajectories.

3.1.1 Dataset Structure

The initial dataset was derived from Wi-Fi traces collected at the venues hosting the 2024 edition of the Sónar Festival. Specifically, it was a log of Wi-Fi activity containing a collection of records where each entry corresponds to a single event involving a device and the network, such as scanning for available Wi-Fi networks or interacting with one. Each record includes the device's MAC address (serving as a unique identifier)*, the event's timestamp, location-related information (e.g., latitude, longitude, H3 cell, floor number), and additional fields that were not essential to the analysis and are therefore omitted here for conciseness.

It is important to note that modern devices commonly implement MAC address randomization, a privacy mechanism that periodically changes the device's MAC address during Wi-Fi scans under certain conditions, making it more difficult to track users across networks [30, 31]. However, it is possible to identify which MAC addresses are randomized by analyzing their received values. Furthermore, for non-randomized cases, device manufacturers or vendors can be determined based on their Organizationally Unique Identifier (OUI) [32].

Given this, the first task of the project was to flag randomized MAC addresses, extract device manufacturers or vendors, and anonymize the received addresses using a hashing function to protect user privacy while preserving valuable information from their unique identifiers. The most relevant fields, derived from this step, are summarized in Tab. 3.1.

Field	**Description**
Anonymized MAC-address	Unique device identifier
Timestamp	Time of the event, as recorded by the Access Point
Lat, Lng, H3 cell, floor num.	Location-related information, including geographic coordinates and location identifiers
Randomized flag	A binary flag indicating whether the MAC address is randomized: 0 if not, 1 if randomized
Device manufacturer/vendor	Device manufacturer/vendor name. It is null for the randomized cases

Table 3.1: Description of the main fields in the dataset.

3.1.2 Analysis Scope and Initial Counts

As Sónar 2024 was held in multipurpose venues with multiple halls, only some predefined areas were designated for the Sónar Festival, as shown in Fig. 1.1. As a measure to prevent unrelated halls' Wi-Fi activity from corrupting the analysis, only the observations from the designated areas were included.

Similarly, both Sónar by Day and Sónar by Night have their own temporal scope. For all days of the festival, Wi-Fi activity was considered from one hour before the first event to one

*Randomized addresses are not guaranteed to be unique, but the large 48-bit space makes collisions in a local network highly unlikely [28, 29] hour after the last event listed in the lineup.

The number of observations and unique MAC address counts resulting from the definition of these scopes are summarized in Table 3.2. It is worth mentioning that attendees could participate in both Sónar by Day and Sónar by Night, depending on the tickets they purchased. As a result, the total number of unique MAC addresses does not match the sum from each individual component of the festival.

	Sónar by Night		Sónar by Day	
	Unique MACs	Observations	Unique MACs	Observations
Non-Randomized	1,446	362,990	3,970	545,393
Randomized	3,014	1,521,114	2,456	3,526,090
Total	4,460	1,884,104	6,426	4,071,483

Table 3.2: Count of unfiltered unique MAC addresses and observations after defining the spatial and temporal scope.

An important aspect to keep in mind is that the headcount measurements announced by the organization yielded a higher attendance count for Sónar by Night (66,000), compared to Sónar by Day (54,000) [33]. This discrepancy might be explained for several reasons.

First, as identified in the previous Sónar paper [4], the dataset included events from devices near the boundaries of the venues, potentially from pedestrians passing by. This phenomenon might be more pronounced for Sónar by Day, as its timetable spanned several daylight hours, while Sónar by Night took place between 20:50 and 7:00 the next morning, isolating it from people's regular activities. Additionally, Sónar by Day included exhibitions, workshops, talks, and forums, some of which were tech-related and may have required devices such as computers, tablets, etc., for their operation, which fall outside the scope of this thesis.

Both datasets were cleaned following the methods described below, resulting in a count of unique MAC addresses that better aligns with the headcount measurements, as shown in Sec. 3.1.7. However, due to Sónar by Night having a larger audience, less noise from the start, and other characteristics that facilitated its treatment (consistent time span over the two nights, focus on music, and being confined to a single floor, thus reducing potential measurement errors), it was chosen for the full analysis. Therefore, from this point forward, all plots and information refer to Sónar by Night, unless otherwise explicitly stated.

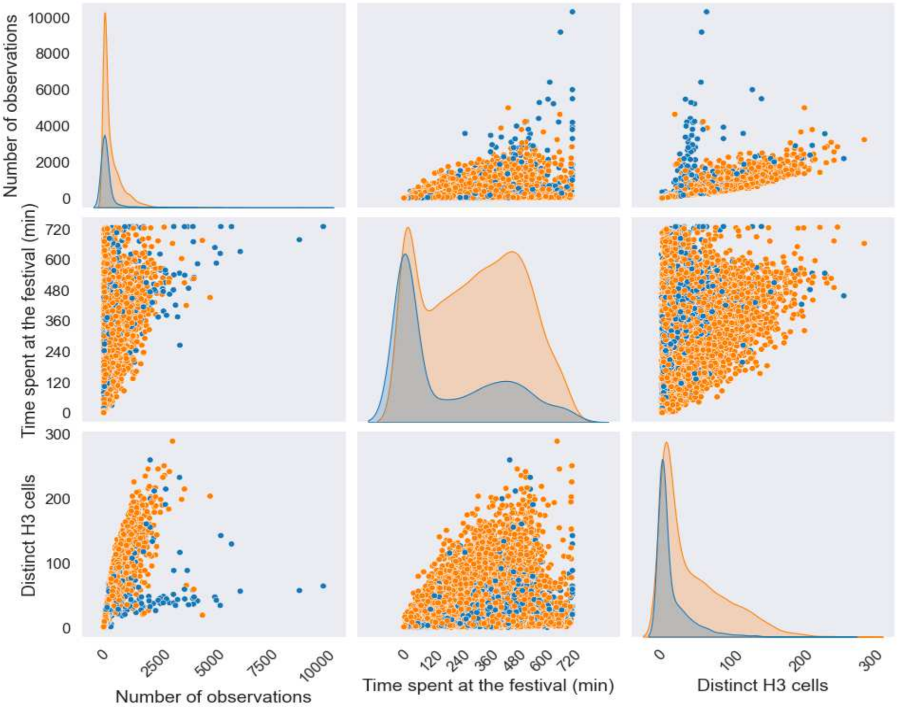


Figure 3.1: Initial distributions of measures by device and night (TID). Non-randomized MAC addresses are shown in blue, and randomized MAC addresses are shown in orange.

3.1.3 Initial Distributions and Single H3 Cell Devices

The combination of the anonymized MAC address and the day/night period effectively forms a unique identifier for each device's activity session. Hence, it will be conceptualized as a Trajectory Identifier (TID) moving forward. As a starting point, three measures were calculated for each TID: the number of observations, the time spent at the festival (computed as the difference between the corresponding maximum and minimum timestamps), and the number of distinct H3 cells[†] where the device was spotted. The initial distributions of these quantities are shown in Fig. 3.1.

An initial observation is that both randomized and non-randomized devices exhibited both short and long durations of time spent at the festival. This could be partly explained because randomized MAC addresses can be consistent with interactions with the same network, at least for a period [34, 31, 35]. Since the lifespan of randomized MAC addresses can distribute over a wide range of durations, including the entire festival, they can be used effectively for the anal-

[†] An H3 cell is a spatial index that partitions the Earth's surface into hierarchical hexagons for geospatial analysis. For this dataset, each cell's area was approximately 43.9 m^2

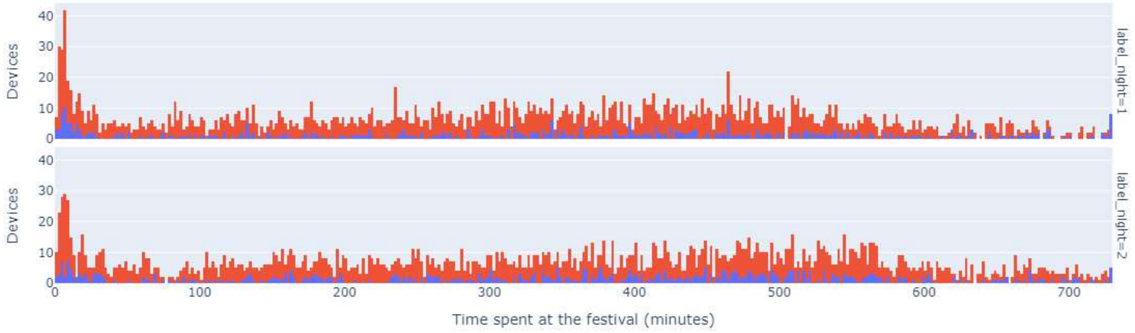


Figure 3.2: Distribution of the time spent at the festival after single cell filtering. Non-randomized MAC addreses are shown in blue, and randomized MAC addresses are shown in red.

ysis.

On a different note, the distributions of the number of distinct H3 cells and the number of observations are highly skewed. Many devices are clustered on the left side of the distributions. A more detailed inspection of these two variables is provided in Sec. 3.1.5. However, an initial examination of the number of distinct H3 cells where the devices were detected revealed that 1,172 TIDs, many of which corresponding to non-randomized addresses, were only observed in a single cell throughout the entire night. These instances included devices with only one observation and others that were detected for extended periods, possibly corresponding to devices fixed by the event organizers or venue. Since these MAC addresses do not provide information about the attendees' behavior at the festival, they were removed from the dataset, leaving 664 non-randomized and 2,736 randomized devices.

3.1.4 Distribution of the Time Spent at the Festival

The time spent at the festival on each night has a particular distribution due to its bimodality. For both nights, it exhibits a strong peak below the 100 minutes threshold and another smaller peak around the 480 minutes mark (8 hours). Fig. 3.2 provides a finer-grained look at this phenomenon after applying the single-cell filter.

A simple remark from Fig. 3.2 is that the rightmost bin of the distribution exhibits a small peak of mostly non-randomized devices (unlike the rest of the dataset). Given that the dataset was bounded by the time when events took place, and the time spent at the festival was calculated by subtracting the minimum observed timestamp from the maximum observed timestamp, this small cluster is likely caused by devices staying more than 720 minutes (12 hours) within the festival's venue—perhaps fixed or organizational devices. As this part of the distri-

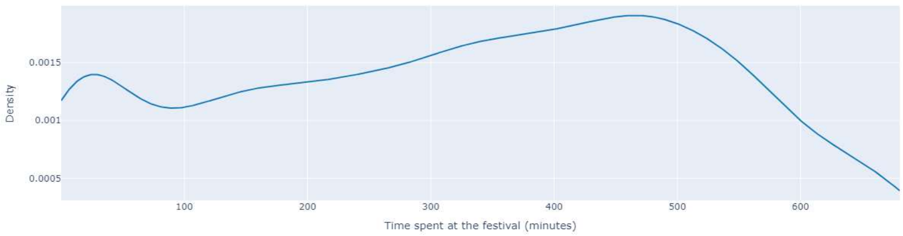


Figure 3.3: Density plot of time spent at the festival per device (Sónar by night). Stays longer than 682 minutes already removed.

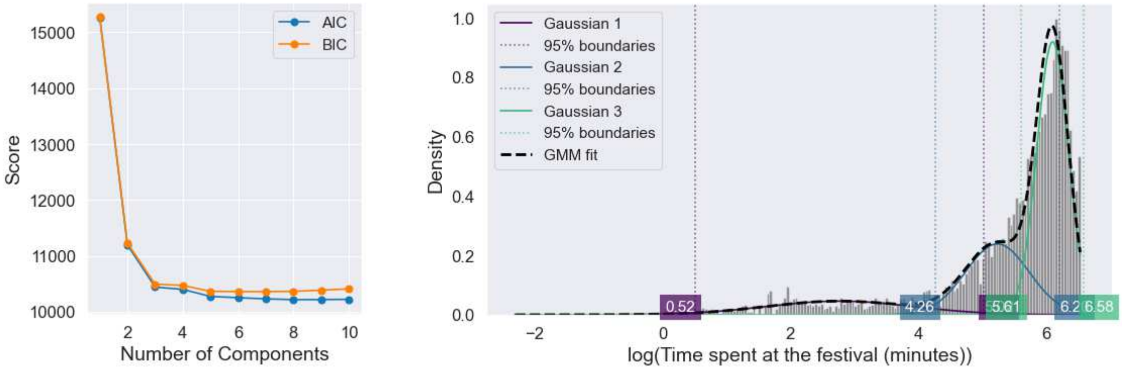
bution refers to durations longer than the total time span of the events for each night (10 hours), an upper bound can be enforced to better isolate devices of interest for this thesis. The $P_{97.5}$ (682 minutes, or 11.4 hours) was chosen as an upper bound for the time spent at the festival as it covers more than the total duration of the events for each night, with the premise of safely reducing noise without losing much data.

3.1.4.1 Lower Bound on the Time Spent at the Festival Using GMM

On the other hand, regarding the concentration on the left side of the distribution, it can be seen that it coincides with TIDs having few observations and distinct H3 cells. This could potentially be caused by people who did not participate in the festival and were simply pedestrians passing near the venue. Due to the limited information and high potential noise that these instances could bring to the analysis, it is desirable to remove them from the dataset. However, deciding on a threshold to remove TIDs that spent a very short time at the festival is not straightforward. Using standard techniques for outlier removal, such as Tukey's method, would be ineffective due to the large concentration of data points in this part of the distribution. Furthermore, there is no obvious value to fix for filtering based on intuition or visual inspection. For this reason, a model-based criterion was chosen to define the threshold value. One possible approach is to assume that the distribution of time spent at the festival is a combination of Gaussian distributions and model the data using Gaussian Mixture Models (GMM). The main idea behind this is to identify different clusters based on the length of stay at the festival and remove the leftmost group of instances, which correspond to devices that spent little time at the festival.

Since the distributions of randomized and non-randomized MAC addresses have similar shapes, and these shapes are consistent across the two nights, it is reasonable to examine this phenomenon as a whole, without splitting the data, as shown in Fig. 3.3.

Although the unaltered distribution already displays a bell-like shape, initial attempts to apply GMM to the raw data did not yield the expected results; the fitted probability density function did not align well with the observed data, and it was not possible to isolate the concentration on the left into a distinct group. Considering this, and recognizing the wide range of values spanning nearly three orders of magnitude, a logarithmic transformation was applied to improve the data's spread. This transformation, which aligns with the natural lower bound of the strictly non-negative durations, reshaped the left concentration into a long tail, better capturing distortions in the distribution and allowing for a better fit and coherent groupings in the GMM results. The number of Gaussian components for the model, determined using the elbow method based on the Bayesian Information Criterion (BIC), was set to three, as illustrated in Fig.3.4.



(a) GMM model selection.

(b) Logarithmic transformation and one-dimensional GMM.

Figure 3.4: GMM on the logarithmic transformation of the distribution of time spent at the festival in a single night.

Fig.3.4b shows that the left tail was effectively encapsulated by a Gaussian component, leaving the two rightmost components isolated from this distortion. The threshold for filtering the devices that spent a very short time at the festival on a single night was determined by taking the 95% intervals of each Gaussian component and removing the instances below the left boundary of the two rightmost components. After reversing the logarithmic transformation and rounding, this yielded 71 minutes. This threshold is reasonable in the context of event durations, as it primarily excludes attendees who did not stay for at least one event, considering that no event was shorter than 50 minutes and the average duration of events was 77 minutes.

The effectiveness of this approach in removing the concentration of devices that spent a very short time at the festival can be seen in Fig. 3.5.

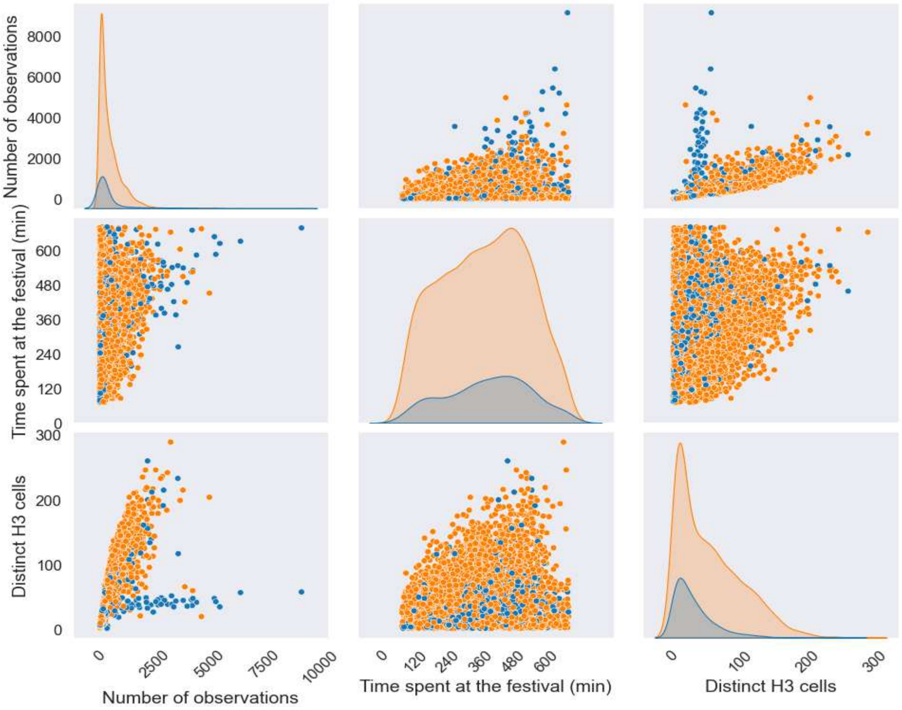


Figure 3.5: Distributions of measures by device and night after the time at the festival filters. Non-randomized MAC addresses are shown in blue, and randomized MAC addresses are shown in orange.

3.1.5 Number of Observations and Distinct H3 Cells per Trajectory

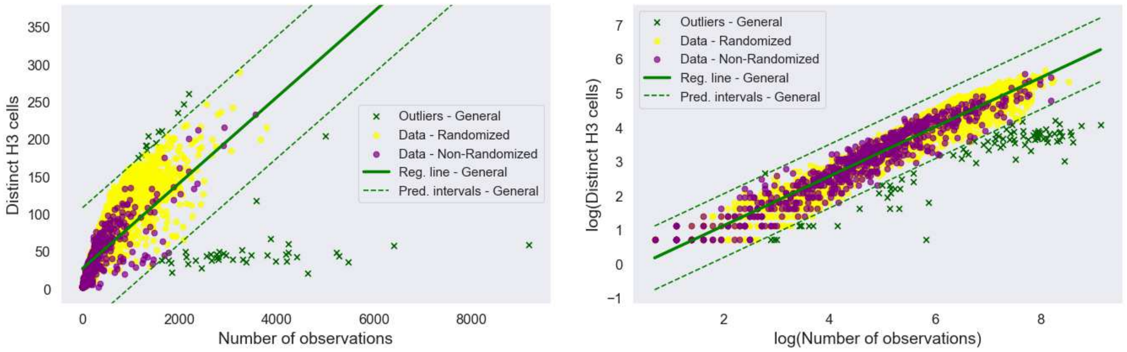
The distributions up to this stage of preprocessing are shown in Fig. 3.5. These plots reveal a positive correlation between the number of observations and the number of distinct H3 cells, in addition to the remaining skewness of the distributions. This correlation is justifiable, as it is expected that devices with many observations were detected in multiple places as they moved.

However, some instances deviate from the trend, as they appear to form a flat line, accumulating many observations without increasing the distinct H3 cell count. This atypical behavior likely corresponds to devices that moved within a small or confined area. It is also plausible that some devices were actually static yet identified in more than one cell due to potential capturing errors. Again, these characteristics possibly point to organizational or venue devices.

To statistically identify these MAC addresses, one strategy is to fit a linear regression model to capture the positive correlation trend. Devices that deviate from this trend can then be flagged as outliers based on the prediction intervals at a given confidence level. Prediction intervals are chosen over confidence intervals in this context because they account for both the uncertainty in the estimated regression line and the variability in individual observations. This broader range helps ensure that fewer data points are flagged as outliers, which is desirable to avoid excessive data loss in the event that they are removed.

An initial approach involved fitting separate regression models for non-randomized and randomized MAC addresses to capture trends specific to each group. However, this strategy proved ineffective due to an apparent bias: most nearly static devices were non-randomized, causing the separate regressions to fail in identifying these devices as outliers. In contrast, fitting a single regression model for all instances captures the overall trend more effectively, enabling the identification of devices that moved very little, regardless of whether their MAC addresses are randomized or not.

On the other hand, the data exhibits heteroscedasticity, as the spread of the data is not constant across the range of the variables. To address this issue, a log-log transformation was applied before fitting the regression, which helped stabilizing the variance and improved the fit quality, with the R^2 increasing from 0.53 to 0.90. The final regression, along with the 99% prediction intervals and the points marked as outliers are illustrated in Fig. 3.6b.



(a) Regression on the non-transformed data. $R^2 = 0.53$.

(b) Regression on the log-log transformation. $R^2 = 0.90$.

Figure 3.6: Linear regression and 99% prediction intervals for the number of observations and the number of distinct cells.

Many of the outliers identified using this method corresponded to non-randomized MAC addresses, so the manufacturers/vendors of these devices were manually examined. This inspection confirmed that many flagged outliers were indeed organizational and non-mobile devices, as the manufacturers were dedicated to the fabrication of POS terminals, Wi-Fi access points, routers, and similar devices. Consequently, all devices from these manufacturers were removed from the dataset. In other cases, it was unclear whether the flagged device manufacturers were related to mobile devices, and there were even some randomized devices, making it impossible to determine the origin of the MAC addresses. Given the effectiveness of the described strategy in detecting non-attendee devices based solely on statistical criteria, the outliers were removed in these ambiguous cases, but other devices from the same manufacturers were kept if they followed the overall trend.

3.1.6 Sparse Trajectories

At this stage, the number of observations and the distinct H3 cell count distributions are well aligned. However, the skewness indicates that the majority of instances have only a few observations, while a handful of instances have significantly more. Using typical outlier removal techniques based on the interquartile range is undesirable, as these methods would aim to remove parts of the distribution corresponding to instances with many observations in a single night. These instances are potentially highly valuable as they offer more information through their higher number of observations.

In contrast, TIDs with few observations may provide limited information on the behavior patterns of attendees, as little is known about their activity during the event. This holds true even if their estimated time at the venue is long, given that the time spent at the festival was calculated using the first and last observation for each night. As observed in Fig. 3.5, TIDs with few observations are dispersed throughout the entire range of the time-span variable, leading to sparse trajectories over the night. This phenomenon might be explained by irregular use of the Wi-Fi network among attendees. Some devices may have interacted with the network more or less frequently due to differences in scanning and connection behaviors, network coverage, or power-saving features, all of which could influence the number of observations in the log.

Due to the wide range of values in the distribution of the number of observations, a logarithmic transformation is useful for visualization, as depicted in Fig. 3.7. Although this plot redistributes the spread of the data, it does not reveal any evident outlier-like behavior at the lower end of the distribution. In any case, these data points should not be dismissed as noise, as there is a plausible explanation for their occurrence. Moreover, given that many trajectories display this behavior, it might not be beneficial to discard this data carelessly, as having more instances may contribute to a more robust understanding of the overall trends.

For this reason, $P_{2.5}$ (9 observations) was chosen to remove only the extremely sparse trajectories, aiming to maximize the use of the data. However, it is worth noting that trajectory sparsity remains a significant challenge for this project and may lead to difficulties later on.

After going through all this preprocessing pipeline, the clean MAC address counts are finally obtained, as shown in Tab. 3.3. Additionally, the updated distributions are depicted in Fig. 3.8.

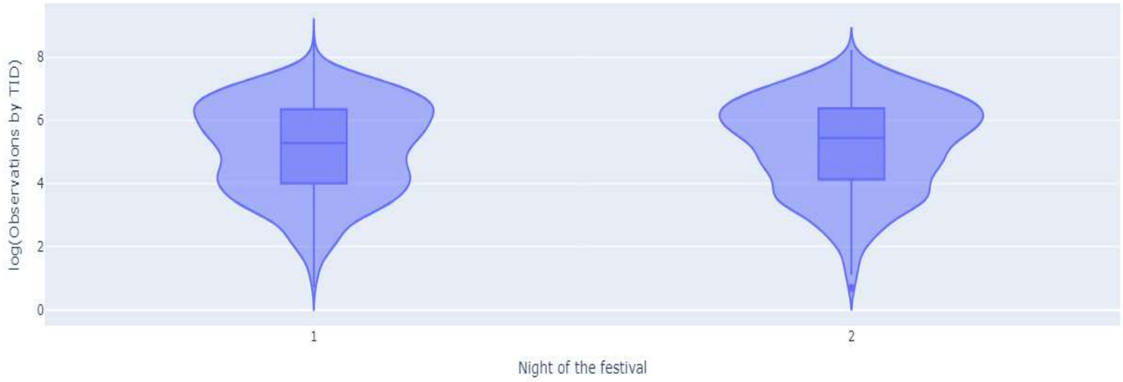


Figure 3.7: $\log$-transformed distribution of observation counts (with previously mentioned filters already applied).

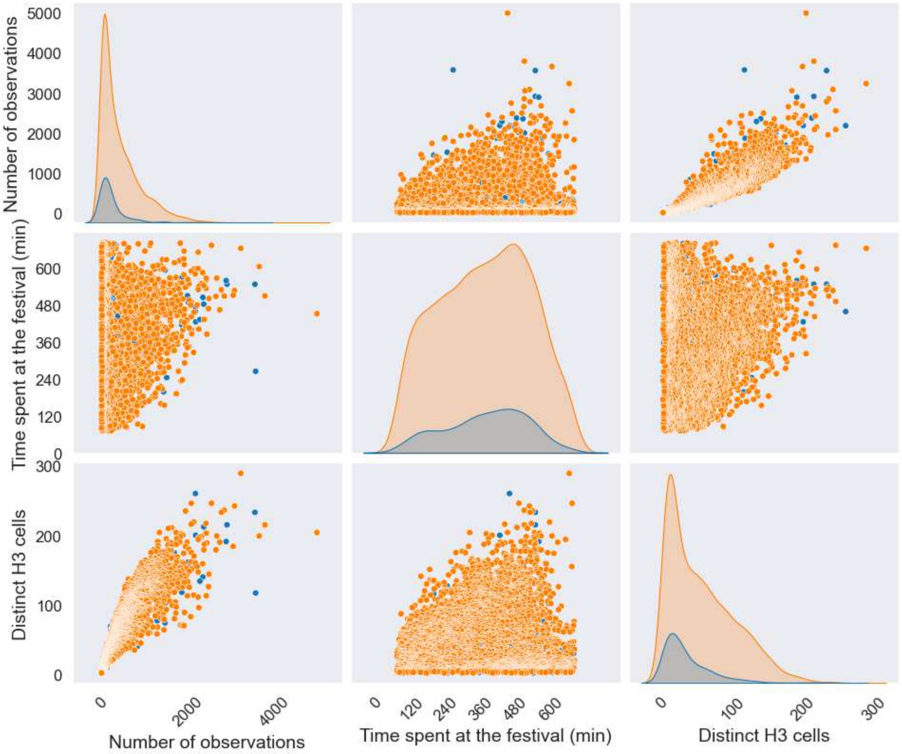


Figure 3.8: Final distributions of measures by device and night (TID) after the cleaning process. Non-randomized MAC addresses are shown in blue, and randomized MAC addresses are shown in orange.

3.1.7 Trajectory Preprocessing and Cleaned Counts

Although no further cleaning steps were applied to remove MAC addresses, some preprocessing procedures were still performed for each trajectory. As trajectory data usually contains positional noise from signal-related errors when recording the data, several preprocessing techniques exist to correct points that deviate from the true path [1].

In this case, trajectory noise was filtered by removing points where the calculated speed between successive points exceeded a specified threshold, which is a common method to remove outlier points from trajectories [1]. The maximum speed used to filter the data points was 10 km/h, which is greater than the average running speed of men [36]. This step removes, for example, observations that occur simultaneously at more than one position, which is a problem that was detected in the dataset. After this step, a small number of exact duplicate observations remained in the dataset (161 for Sónar by Night) and were later removed.

Finally, the resulting counts after the complete cleaning pipeline are shown in Tab. 3.3.

	Sónar by Night		Sónar by Day	
	Unique MACs	Observations	Unique MACs	Observations
Non-Randomized	453	134,850	769	169,841
Randomized	2,342	1,109,611	1,985	2,176,154
Total	2,795	1,244,461	2,754	2,345,995

Table 3.3: Count of filtered unique MAC addresses and observations after the cleaning process.

As a side note, the same overall procedure (with minor adaptations for the specific case) was followed to clean the Sónar by Day dataset, although it was not analyzed in this thesis. However, it is still worth noting that the final counts demonstrate that the cleaning techniques are effective and reasonable, as the number of MAC addresses is now more consistent with the actual headcounts reported by the organization, as mentioned in Sec. 3.1.2. Additionally, the higher counts of cleaned unique MAC addresses for Sónar by Night further support the decision to focus the analysis on this modality.

The cleaned dataset was used to generate descriptive plots, shown in Figures A.1, A.2, A.3 and A.4 of the Appendix. These plots provide an overview of the aggregated attendance patterns and the event lineup. Key takeaways relevant to this study include that SonarClub, SonarPub, and SonarLab consistently had the highest volume of observed MAC addresses, and that the temporal distribution of the lineup format and attendee volume is similar for both nights.

3.2 Feature Preparation for Modeling

While the cleaning steps mentioned above effectively remove noise from irrelevant devices and glitchy signals, treating the trajectories further can yield representations with richer and more meaningful content than the complete noise-filtered trajectories alone.

First, each trajectory point was spatially joined to its corresponding delimited area within the venue. The event timetable and the points' timestamps were then used to associate observations with specific events. This process also identified observations not linked to any event, either due to the area's nature (e.g., restaurant zones, entrances) or because no event was occurring at the time of observation.

Moving forward, two forms of tables were derived to support the following stages of the analysis: the sequences of stops, and the general trajectory measures.

3.2.1 Sequences of Stops

As noted in Sec. 2.2, not all recorded points are essential to describe a trajectory. Therefore, the notion of a stay point, represented as $\dot{s}$, plays a crucial role. However, in this case, the usual definition of a stay point, with a fixed spatial radius and temporal threshold, is not ideal.

In this setting, a "stop" at a given location is not defined by a fixed radius. Instead, with respect to the spatial aspect of a stay point, it is more accurately characterized by contiguous observations within delimited areas that can vary in shape and size. Moreover, the nature of these zones (e.g., audience zones, restaurant areas, bars) can inherently suggest the reason for a stay at a given location.

For this reason, a custom sequence of stops was devised and implemented for each trajectory, replacing the typical concept of stay points. The approach used to derive these sequences represents a core aspect of this thesis, as the decisions made at this stage significantly influence the subsequent results.

The main idea was to identify whenever the users' position meant a change of spatiotemporal context, whether it was the delimited venue area or the event (if a user stayed at the same stage for different events, different information could be extracted from each moment). In this context, a "stage" refers broadly to the audience zone associated with a specific performance area, but also includes other large delimited locations, such as the restaurant zone. To ensure clarity and precision, the calculation of the sequence of stops for a single trajectory can be broken down into the following aspects:

1. **Detecting stage or event transitions and assigning stop IDs**: A transition flag is set whenever there is a switch of stage or event between consecutive observations. The cumulative sum of these flags assigns unique stop IDs.

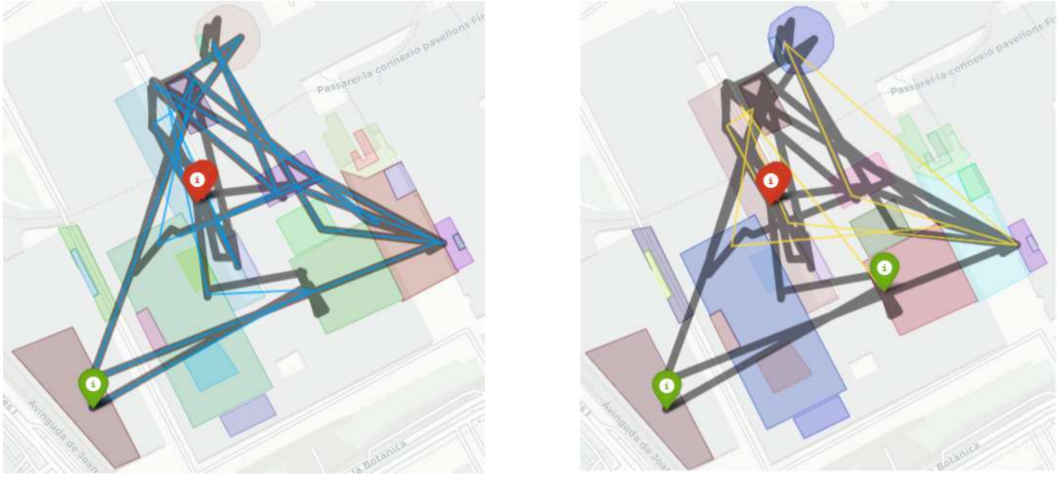
2. **Defining the start time of the stop**:
   - If the stop corresponds to a location switch, the start time is set to the first observation at the new location.
   - If the stop is detected due to an event switch at the same stage, the start time is set to the start time of the new event.

3. **Defining the end time of the stop**:
   - If the stop corresponds to a location switch, the end time is set to the timestamp of the last observation before the switch.
   - If the stop corresponds to an event change within the same location, the end time is set to the end time of the event.

4. **Extracting stop's information**: Each stop ID inherently corresponds to a specific event and stage, while the start and end times define the stop's duration. Additionally, all observations within a stop are converted to UTM coordinates—allowing computations on a local plane—and used to compute the stop's centroid. As a side note, some locations feature different types of areas (e.g., stage or bar zones). For each stop, it was possible to calculate the time spent in each type of area by defining "substops" and aggregating durations accordingly.

This strategy assumes that when consecutive observations occur within the same area, the attendee has not changed locations. It is important to note that for 95% of the data, the time difference between consecutive trajectory points is less than two minutes. This implies that, although there are many sparse trajectories, the majority of observations belong to trajectories with a fine temporal granularity, making this assumption generally reliable. Furthermore, this premise allows for the computation of stop durations that more accurately reflect the temporal context of events. Regarding geographical information, the centroid of the defined stops may have a coarser spatial granularity compared to a traditional stay point with a fixed radius (if the radius is small). However, this more flexible method ensures alignment of points with their correct spatial context within the festival setting.

By extracting the sequence of stops, redundant observations within a given location are consolidated into a single point. This approach reduces the dataset while maintaining the overall


Passarel·la connexió pavellons Fi
Avinguda de Joan
la Botanica
Passarel·la connexió pavellons Fi
Avinguda de Joan
Botanica

(a) No temporal threshold

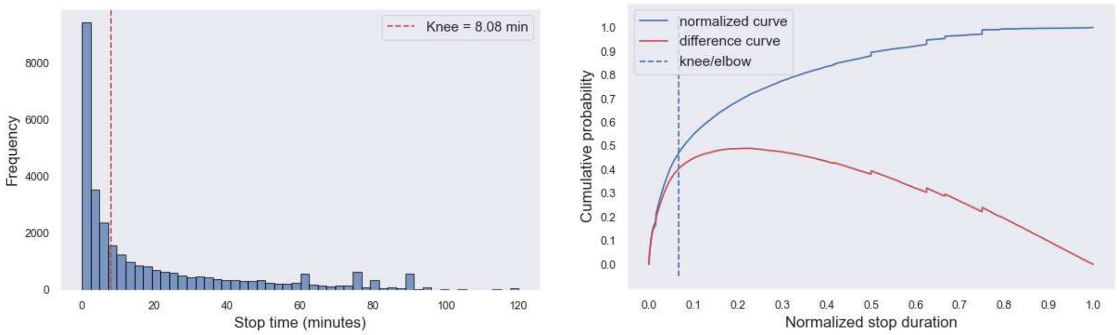
(b) Temporal threshold applied

Figure 3.9: Single example of the sequence of stops (colored lines) and its original trajectory (black shadow). The green and red markers indicate the start and end of each trajectory/sequence.

structure of the trajectories, as illustrated in Fig. 3.9a. However, this process does not explicitly account for the temporal threshold that defines a stay point yet. Consequently, even after this reduction, many trajectory points may still represent transient activities, such as movements between stages or brief pauses, rather than significant stays within an area. To address this, defining a temporal criterion is crucial for isolating meaningful stops.

With this in mind, it is informative to look at the stop time distribution depicted in Fig. 3.10a, which highlights a high concentration of short durations, particularly below the three-minute mark. As there appears to be an abrupt break in the left part of the distribution of stop durations, this suggests a natural threshold for filtering out non-significant stops and retaining only more meaningful and intentional stays at a location. One strategy to identify this break is to compute the empirical cumulative distribution function (ECDF) of stop durations and apply a knee-detection algorithm. The ECDF provides a non-parametric view of the distribution, showing the proportion of stops that fall below a given duration. The knee point, illustrated in Fig. 3.10b and fixed at 8.08 minutes, marks a transition from a steep increase (where many short stops accumulate rapidly) to a more gradual slope, where stop durations become more evenly spread out rather than being dominated by very short stays. This methodology yields a data-driven method for finding the threshold mentioned above. An example of the resulting sequence of stops after applying the temporal threshold is shown in Fig. 3.9b.

This method simplifies dense and potentially noisy signals into compact sequences that pre-



(a) Distribution of stop durations.

(b) Knee point detection from the ECDF of stop durations.

Figure 3.10: Distribution of stop durations and temporal threshold. This distribution excludes stops unrelated to an event, as a few instances exceeding 120 minutes distorted the plots.

serve key information from the original trajectories. By departing from the traditional definition of stay points while maintaining and adjusting their underlying intent, this approach offers versatility. It can be helpful in other scenarios where predefined zones determine the spatial context of stops, as opposed to relying on a spatial radius, which may be less precise and hard to determine.

Due to their sparsity, 159 trajectories without any stops exceeding the temporal threshold were identified. As the Wi-Fi traces for these instances lacked meaningful information for the analysis, they were excluded from the dataset. In addition to this, there were 412 instances where none of their filtered stops corresponded to an ongoing event, likely sparse trajectories, or staff members who tended to remain in a specific location. As these cases are not representative of relevant movement patterns or interactions with the events, they were also discarded from the posterior analyses. This resulted in 3,275 TIDs (combinations of MAC address and night identifier) distributed in the two nights as shown in Tab. 3.4. **These were the final trajectories used for modeling.**

	Unique MACs	Full trajectory observations	Total stop observations (temporal threshold applied)
Night 1	1,580	4,681,560	9,564
Night 2	1,695	5,691,644	10,864
Total	3,275	10,353,322	20,428

Table 3.4: Final trajectory counts used for modeling.

3.2.2 General Trajectory Measures

Apart from the sequence of stops, some aggregated measures were obtained to describe the general aspects of each trajectory. Some are straightforward to describe, while others need a more detailed explanation.

- **Arrival time to the festival in minutes:** The time difference (in minutes) between the festival's start and the first recorded timestamp of the complete trajectory.

- **Departure time from the festival in minutes:** The time difference (in minutes) between the festival's start and the last recorded timestamp of the trajectory.

- **Time spent at the festival:** The time difference (in minutes) between the recorded departure and arrival times.

- **Number of attended events:** Number of different events extracted from the filtered sequences of stops. This way, only significant attendances to events are included.

- **Average H3 cell density:** This metric serves as a proxy for estimating the density of areas visited by attendees, acknowledging that the density in the real world fluctuates continuously and is difficult to measure precisely. It calculates density using the count of devices in H3 cells within 5-minute time windows, with each trajectory point matched to the H3 counts from its respective time window. These values are then aggregated to determine the overall average for the trajectory.

- **Distance as a straight line:** It is a common human mobility metric, computed (in kilometers) with scikit-mobility [37]. The straight line distance d_{SL} traveled by an individual u is defined as the sum of the distances between consecutive trajectory points:

  $$d_{SL} = \sum_{j=2}^{n_u} dist\left(r_{j-1}, r_j\right), \tag{3.1}$$

  where n_u is the number of points in u's trajectory, r_{j-1} and r_j are two consecutive points from the time-ordered complete trajectory, and $dist$ is the geographic distance.

- **Radius of gyration:** Another common metric used in mobility analysis [2, 37] to indicate the *characteristic distance* traveled by u, defined as:

$$r_g = \sqrt{\frac{1}{n_u}\sum_{i=1}^{n_u} dist\left(r_i, r_{cm}\right)^2}, \tag{3.2}$$

where r_i represents each trajectory point, and r_{cm} is the center of mass of u's trajectory. It indicates the spatial spread of movement. It was also obtained in kilometers with scikit-mobility.

- **Total stops' duration:** The total sum of stops in the filtered sequence. It is different from the total time spent at the festival, as there can be time gaps between stops. This reflects the actual "observed" time for each trajectory.

- **Proportion of total stops' duration without ongoing events:** The proportion of time spent at stops where events were not occurring, relative to the total stops' duration.

- **Proportion of total stops' duration in each location type:** The proportion of stops' duration across different location types. There were three types of locations: spectator areas, bar/restaurant zones, and other areas (e.g., entrance, bumper car zones). The stages—here defined as the general audience zones associated with performance areas—may include both spectator areas and bar areas, which is why these two categories are distinguished.

- **Score associated with the attendance to each event:** Instead of the raw durations at each event, there is an alternative metric that can be more informative with respect to the musical preference of attendees. In the previous study based on the Sónar Festival data, the authors devised a score to measure the *implicit feedback* provided by an attendee to an event through the time they spent there in relation to their overall participation patterns throughout the festival [4]. Apart from the duration at each event, this score incorporates the total time spent across all events for a given attendee, as well as certain correcting factors. Specifically, this *adjusted score* can be expressed as:

$$AdjScore_{u,e} = \frac{1}{\sum_{\tilde{e}\in s(e)} dur_{u,\tilde{e}}}\left(\frac{dur_{u,e}}{\sum_{e\in E} dur_{u,e}}\right)\log\left(\frac{dur_e}{l_e \cdot c_{s,e}}\right), \tag{3.3}$$

where $dur_{u,e}$ is the duration of user u at event e, E is the set of events and $\sum_{\tilde{e}\in s(e)} dur_{u,\tilde{e}}$ represents the total time user u spent in the stage s where event e occurred. This last expression was included to penalize a bias caused by attendees spending an extended time at a specific location, regardless of the event taking place. On the other hand, the element on the right, log $\left(\frac{dur_e}{l_e \cdot c_{s,e}}\right)$, corresponds to a "general factor" that adjusts each score based on the following:

- The total time spent by all users at event e, dur_e.
- The duration allocated for the event e in the lineup, l_e.
- The capacity of the stage s where event e occurred, $c_{s,e}$. In this case, the area of the designated polygon was used as a proxy.

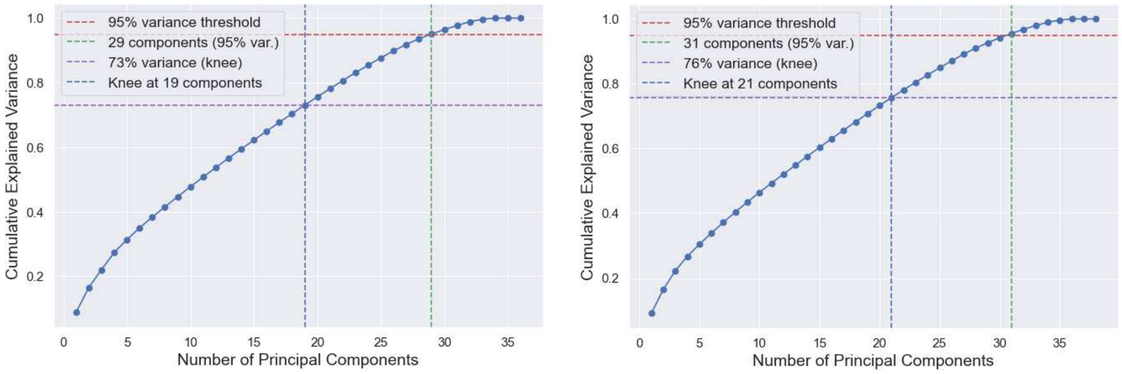
Units of some measures are explicitly mentioned for the sake of clarity and interpretability. For example, the arrival and departure times were converted from the raw timestamp to relative minutes for a clearer interpretation when analyzing the nightly results. However, when preparing the data for the clustering algorithms, these features were z-normalized to prevent any of them from dominating the clustering.

Another important remark is that there was an attempt to group artists by musical style to obtain more aggregated scores. However, the festival has an experimental nature and some artists are difficult to classify solely on the basis of the information available online. Moreover, at least for this night edition, most of the events featured different types of electronic music, with only a few acts diverging from these styles. As a result, the groupings of *artists* were rather homogeneous, unbalanced, and hard to determine objectively. For this reason, the individual scores for each event were ultimately used.

Since there are multiple acts for each night (24 for night 1 and 26 for night 2), including the scores for all events in addition to the other general measures increases the dimensionality of the problem, potentially including non-relevant features for clustering (which is an issue further discussed in Sec. 3.3.4). To evaluate this potential issue, a Principal Component Analysis (PCA) was performed on the matrix of general features to evaluate how many components explain the variability in the data. PCA is a linear dimensionality reduction technique that can help assess the intrinsic dimensionality of the data by transforming the original features into new, uncorrelated variables (principal components) ordered by the amount of variance they capture [7]. If only a few components explain most of the variance, the data can be effectively reduced in dimensionality without significant information loss.

However, PCA showed that both nights required nearly all components to explain most of the variance (Fig. 3.11) in the general measures, indicating that many dimensions are needed to capture the variability in the data—at least in the context of the linear projections provided by

PCA. Moreover, applying the knee method to reduce the number of projected features would result in a loss of about 25% of the variance in the data, as the plot does not exhibit a steep initial rise. Therefore, all described measures were retained for clustering and later analyzed within the resulting clusters to assess their role in characterizing each group. In any case, the dimensionality of the formulation was later addressed with an alternative non-linear projection method, discussed in Sec. 3.3.5.



(a) PCA for night 1.

(b) PCA for night 2.

Figure 3.11: Variance explained with PCA for the general measures of each night.

3.3 Modeling Frameworks

Building on the features obtained for analysis, this work's dissimilarity-based clustering framework comprises two main components: the distances used to measure dissimilarity between attendees and their trajectories, and the clustering algorithm implemented to detect the groupings. This section outlines the specific methodology employed, including implementation choices and adaptations based on the theoretical principles introduced in Section 2.

With respect to the dissimilarity measures, the general approach was to use a combination of two distances: a *general features distance*, D_{GF}; and a *time series distance*, D_{TS}, which was borrowed from one of the distances described in detail in Sec. 2.3. The two metrics are mixed with a λ coefficient that controls the influence of the time series distance over the final measure:

$$D_{combined} = D_{GF} + \lambda D_{TS}. \tag{3.4}$$

To prevent either measure from disproportionately influencing the overall result, both distances were normalized to the $[0, 1]$ range using the min-max scaling method. The idea behind this approach is to integrate the summarized information, derived from the complete attendees' presence at the festival, with the finer spatial and temporal nuances that the time series might bring to the analysis. The variable λ offers flexibility in examining to what extent the time series distance influences the formation of clusters in the data. Meanwhile, fixing D_{GF} helps to provide context to the combined distance and keeps the analysis grounded and interpretable.

The general features distance, D_{GF}, is simply the Euclidean distance in a space formed by the normalized features outlined in Sec. 3.2.2, while the specific implementations of each of the time series distances are described in more detail in Sections 3.3.1, 3.3.2, and 3.3.3. In addition, Sections 3.3.4 and 3.3.5 address important aspects of clustering algorithms and data representation after obtaining the combined distances.

As a safety measure for consistency in analyzing the time series, the two nights were analyzed separately since each night had a unique context, making it better suited for drawing conclusions from the results.

3.3.1 DTW and D_{GF}

In this case, the implementation was direct, as the pairwise D_{TS} was simply the standard DTW distance (see Sec.2.3.1 for details) applied to the spatial coordinates of the sequences of stops. For this approach, the core work was already done when obtaining and filtering the sequences, since the resulting *silhouettes* of meaningful stops, derived from this process, determine the similarities of the overall trajectories. This approach focuses on the general movement patterns of attendees, allowing time offsets and adding richer context with the general features distance, D_{GF}.

3.3.2 LCSS and D_{GF}

In this work, the approach for implementing LCSS was different. As described in Sec. 2.3.2, the LCSS for trajectories usually requires parameters ϵ and γ, which are spatial and temporal thresholds for matching points in two sequences.

Exploratory attempts were made to fix ϵ for the sequences of stops, but determining a suitable value was nontrivial given the festival's layout, which consists of delimited areas with varying shape and size. Similarly, for γ (which is not usually considered explicitly in public implementations), various values were tested, interpreting it as an index-based constraint (sequence order) and as a constraint in *physical time* (actual stop duration). Like ϵ, there is no straightforward method to define a suitable γ for time tolerance in this context.

To address these difficulties and building upon the sequence of stops, the LCSS was adapted to align with the festival events, which are inherently connected to the meaningful temporal and spatial context of attendees' movements. Specifically, each event was assigned a unique integer and ϵ was set to 0, effectively treating event labels as categorical variables and requiring exact matches. Consequently, two trajectory points were matched within the LCSS framework only if attendees participated in the same event. Since events have predefined start and end times, an explicit γ constraint was unnecessary.

Compared to the combined distance involving DTW, this approach places less emphasis on the spatial distribution of trajectories, as all points within a stage during an event are summarized into a single category. While this results in a coarser geographic granularity, it aims to align trajectories more explicitly with the festival's lineup. It is also important to note that LCSS tolerates breaks within the alignment. Given the small set of possibilities (with relatively few stages and events to choose from), this framework may lead to certain inconvenient effects, as discussed in Chapter 4.

3.3.3 CTWE AND D_{GF}

As an alternative to DTW and LCSS, a method referred to as Contemporary Time Window Euclidean (CTWE) distance was devised and implemented with the specific music festival setting in mind. Unlike the previously presented time series dissimilarity measures, this approach disallowed flexibility in time alignment, focusing instead on enforcing contemporaneous similarities in trajectories, following the intent described in Sec.2.3.3 with the *average contemporary distance*. The aim was to explore whether clustering attendees' movement patterns could benefit from this stricter alignment approach.

However, the *average contemporary distance* is based on a shared temporal domain. This condition does not apply in this context, as participants could enter and leave the festival at different times, and the sample rate of observations was not consistent across trajectories. This made it impossible to measure the distance between contemporary observations. To address this issue, the sequences of stops were partitioned into time windows, with the time spent at each zone aggregated for each time window. In this way, a sequence of vectors was obtained, where each vector of dimension d represents the time spent at each location for the given time window. The dissimilarity between these time series was then measured using the standard Euclidean distance (Eq. 2.4). For two sequences A and B, segmented into TW time windows:

$$D_{CTWE}(A,B) = \frac{1}{TW}\sqrt{\delta(a_1,b_1)^2 + \ldots + \delta(a_{TW},b_{TW})^2}, \tag{3.5}$$

Dividing by TW scales the result, which is not particularly relevant here since all sequences share the same length, but may be useful when comparing different time window lengths. This approach enforces a coarse temporal alignment between trajectories but lacks spatial and temporal granularity with respect to the other alternatives, as it only considers aggregated measures.

Two time window lengths were initially considered: 15-minute windows and broader windows aligned with different concert phases, identified based on the lineup and abrupt shifts in crowd distribution. Each night was divided into seven time windows. As this formulation is essentially equivalent to flattening the vector sequences into a single array with a dimension of $TW \cdot d$ for each trajectory, exploratory runs showed that the 15-minute windows performed worse, likely due to the curse of dimensionality (discussed in the following sections). Consequently, the broader windows were chosen.

Additionally, to carefully assess CTWE's performance under dimensionality reduction, a PCA version was implemented on the flattened vectors. The cumulative variance behavior was found to be similar to that of the general features matrix (Fig.3.11), characterized by a gradual increase and the requirement for many components to explain most of the variance. To evaluate the effects of significantly reducing the dimensionality of CTWE, the PCA version based on the knee point of the cumulative variance plot was selected for each night, explaining 71% and 70% of the variance, respectively.

3.3.4 Clustering algorithms

Since the combined distance, $D_{combined}$, is a custom measure, its treatment and representation are not straightforward, as its calculation does not originate from a predefined space. For this reason, K-medoids and density-based clustering techniques, such as HDBSCAN, are good candidates for this task, as they do not rely on the computation of cluster means and support custom distortion measures, as described in Sec. 2.1. Furthermore, testing two different approaches (partitioning and density-based) allows for the evaluation of their respective effectiveness in grouping the data in this setting. Lastly, from a practical standpoint, HDBSCAN simplifies the hyperparameter search compared to standard DBSCAN, while K-medoids requires only the number of clusters, K, to be specified in advance. This is particularly convenient since λ is an additional variable to consider in the search space for this formulation.

However, after some initial runs, some complications came from trying to obtain the clusters directly from the combined distances. First, for multiple combinations of hyperparameters, HDBSCAN had a tendency to classify many samples as noise, limiting the usefulness of its output. This phenomenon might be explained because the combined distance integrates global measures and time-series features (each of which can involve multiple dimensions) which increases the dimensionality of the feature space, potentially leading to the curse of dimensionality, where distances lose discriminative power and clustering methods struggle to identify dense regions [38].

Additionally, evaluating the effectiveness of clustering was not straightforward as, in the absence of a ground truth, it was necessary to rely on internal metrics that measure the ratio of within-cluster scattering to between-cluster separation, such as the Silhouette score. The issue in this case was that these metrics are typically better suited for globular clusters and are not specifically designed to validate density-based clustering techniques like HDBSCAN, where arbitrarily shaped groups can form and noise is a possible outcome. For this reason, a measure called the Density-Based Clustering Validation index (DBCV) is more appropriate [39], as it evaluates the clustering structure based on density and connectivity. However, DBCV was originally formulated assuming Euclidean or Squared Euclidean distances, as certain properties and definitions rely on these assumptions. Additionally, most publicly available implementations of DBCV require point coordinates rather than a precomputed distance matrix, making its application more straightforward when working in a vector space. As the formulation proposed in this thesis involves a complex measure that deviates from the standard Euclidean distance in some cases, this introduced challenges in terms of both theoretical validity and practical implementation.

Lastly, since $D_{combined}$ is a custom distance, it does not correspond to an explicit vector space representation, making it difficult to visually assess how clusters are organized. For this type of task, it is desirable to have visual feedback, as it can intuitively guide the implementation process.

To address these issues, data was projected using the Uniform Manifold Approximation and Projection (UMAP) [40].

3.3.5 Projection using UMAP

UMAP is a non-linear dimensionality reduction technique that creates a lower-dimensional representation of the data while preserving its local structure. It models the original data as a high-dimensional graph and optimizes a similar graph in a lower-dimensional space, aiming to maintain nearby points close together while also capturing some broader relationships. Compared to methods such as Multidimensional Scaling (MDS) or t-distributed Stochastic Neighbor Embedding (t-SNE), UMAP offers faster computation and mediates the balance between local and global structures [40]. In the context of this work, UMAP helps transform the complex and abstract custom distance into a vector space where clustering and validation can be more easily implemented.

Although clustering in the embedded space produced by UMAP should be approached with caution, as its output does not fully preserve density and may create spurious clusters, the qualities of this technique can still be beneficial for effective and reliable clustering [38]. As with any dimensionality reduction technique, some properties of the original data may be lost or distorted when the information is compressed into fewer dimensions. However, key aspects of the data structure should be retained, while helping to address the difficulties described above.

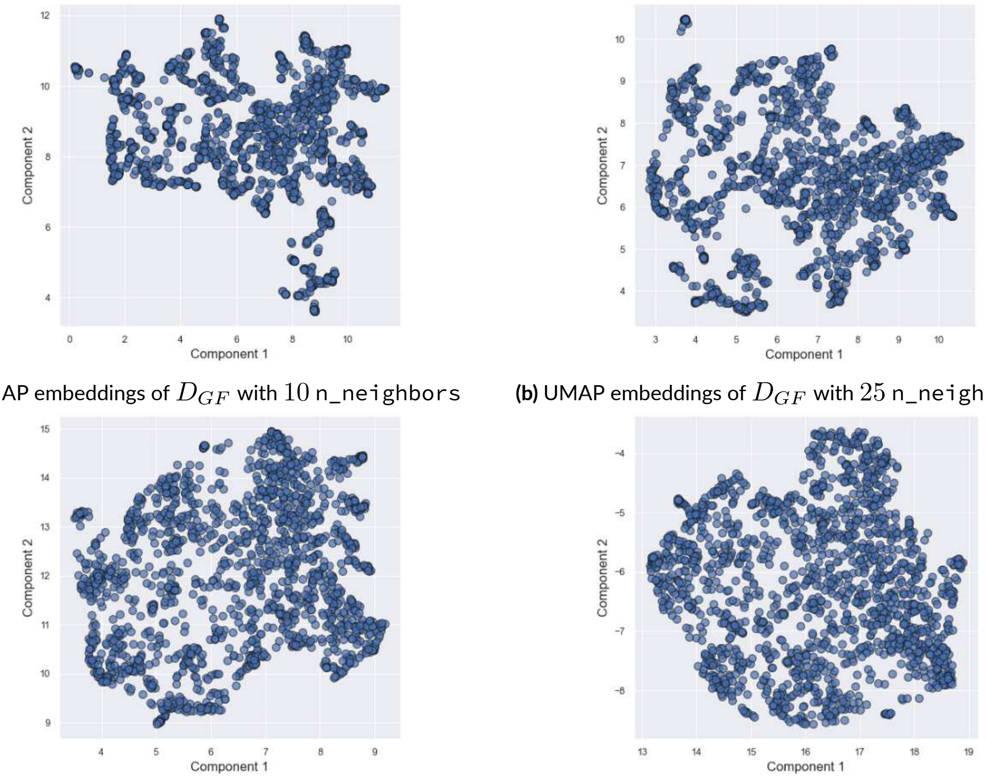
In any case, the resulting clusters from any approach should be examined to assess the reliability of the results. In this particular setting, with no ground truth and an emphasis on interpretability, a qualitative evaluation of the clustering was key to validate the proposed methodology. In this work, UMAP proved to be an effective tool for overcoming practical challenges and extracting meaningful clusters.

It is important to mention that UMAP is widely used for visualization, so projections are typically made in two or three dimensions. However, data can be projected to higher-dimensional spaces for clustering. In this case, to avoid overcompressing the data, two different projections were made:

- A 2D projection used only for visualization.
- A slightly higher-dimensional projection (5D) used for clustering. Since this problem lacks ground truth, assessing the "correctness" of the dimensionality is challenging. Comparing distance- or density-based metrics across different dimensionalities is unreliable because each dimensionality reshapes the space, altering distance distributions and density relationships. Therefore, to establish a consistent basis for tuning other hyperparameters during the experimental phase, the number of dimensions was fixed to provide a more expressive representation than 2D embeddings while keeping the dimensionality relatively low. In practice, this setup led to meaningful results.

UMAP has two other important parameters, `min_dist` and `n_neighbors`. Following the guidelines for clustering with UMAP [38], `min_dist` was set to 0, as it controls how close the points in the embedded space are allowed to be. Low values of this parameter form tightly packed structures, which is beneficial for clustering.

On the other hand, n_neighbors controls how the algorithm balances the emphasis between local and global structures in the data [40]. The UMAP documentation states that a value of 10 n_neighbors should work for most datasets, but it is also recommended to try with "larger" values for clustering, as this helps capture a more global structure and prevents overfitting to local noise, which could lead to overly fine-grained clusters that lack significance [41, 38]. In this case, to examine the effect of this parameter on the data representation, a list of values spread on a logarithmic scale from 10 to a quarter of the number of instances was tested and visually assessed in the 2D projections. For instance, Fig. 3.12 displays some of the UMAP outputs for an arbitrarily chosen distance (the D_{GF}) with fixed parameters, while varying the value of n_neighbors.



(a) UMAP embeddings of D_{GF} with 10 n_neighbors

(b) UMAP embeddings of D_{GF} with 25 n_neighbors

(c) UMAP embeddings of D_{GF} with 65 n_neighbors

(d) UMAP embeddings of D_{GF} with 166 n_neighbors

Figure 3.12: Visual assessment of the n_neighbors parameter for UMAP.

For this setting, it appears that with more than 25 n_neighbors, the local structures tend to merge into a single mass, which is undesirable for clustering. This behavior was observed consistently across all distance configurations and for both festival nights. To preserve the local structure while balancing the trade-off between local and global representations, as recommended when using UMAP for clustering [38], the parameter was fixed at 25 for all distance types and λ values. This approach provided a common ground for tuning the remaining hyperparameters through experimentation, preventing an overly large search space and avoiding reliance on embedding configurations to artificially enhance clustering performance.

3.4 Experimental setup

As discussed in Sec. 3.3, the variable λ, which regulates the role of the time series distance in the final measure, is a key consideration in the proposed formulations. To analyze its impact, λ was systematically varied over $\{0, 0.25, 0.5, 0.75, 1\}$.

On the other hand, for the clustering algorithms, hyperparameter selection was performed, using grid search by varying the values of $MinClusterSize$ and $MinPts$ for HDBSCAN, and K for K-medoids. $MinClusterSize$ was selected from a range of values between 1% and 5% of the total data points, distributed on a logarithmic scale. Additionally, $MinPts$ was defined as proportions of $MinClusterSize$, with the following values tested: $0.16, 0.33, 0.66$, 0.83, and 1. For K-medoids, K ranged from 2 to 10 clusters to test different configurations.

Regarding the validation procedure, DBCV and Silhouette score were used as evaluation metrics, keeping in mind that the DBCV is better suited for HDBSCAN as it is better at handling noise and validating density-based clustering configurations. However, in the absence of ground truth, validating clustering outcomes becomes inherently challenging, as no definitive measure exists to assess the quality of the results.

For instance, during the initial stages of evaluation, it was observed in some cases that the algorithms produced high validation metric values in two problematic scenarios: configurations with numerous small clusters, which likely overfit the data and provided too few observations to draw robust conclusions; or configurations with a single large cluster and a few outliers, limiting the relevance of the results due to the lack of segmentation. In this case, where interpretation is key, achieving well-separated and relatively balanced clusters contributes to making the clustering results more meaningful and generalizable.

To systematically select potentially valuable results for subsequent manual inspection, an additional tailored metric, referred to as the Balanced Clustering Score (BCS), was implemented to evaluate clustering quality. The BCS integrates two aspects: clustering structure, represented by $\Psi(\mathcal{C})$, and cluster size balance, represented by the normalized entropy $\eta(\mathcal{C})$, which yields values between 0 and 1. The computation is defined as follows:

$$BCS = \beta \cdot \Psi(\mathcal{C}) + (1 - \beta) \cdot \eta(\mathcal{C}), \tag{3.6}$$

where $\mathcal{C}$ denotes the clustering configuration, $\Psi(\mathcal{C})$ corresponds to the rescaled DBCV score for HDBSCAN or the rescaled Silhouette score for K-medoids, and $\eta(\mathcal{C})$ quantifies the balance of cluster sizes. Both the Silhouette score and the DBCV typically yield values in the range $[-1, 1]$, so they were rescaled to $[0, 1]$ to ensure consistency in formulation and ease of interpretation. This rescaling guarantees that the BCS also falls within the $[0, 1]$ range. The normalized entropy $\eta(\mathcal{C})$ is defined as [42]:

$$\eta(\mathcal{C}) = \frac{-\sum_{i=1}^{k} p_i \log(p_i)}{\log(k)}, \tag{3.7}$$

where p_i is the proportion of points in cluster i, and k is the total number of clusters. This formulation provides a simple way to combine the compactness and balance of the clusters. This metric was implemented as a practical tool to help examine cluster configurations, avoiding the need for exhaustive manual inspection. However, it should be used with caution, as high entropy values can be achieved simply by assigning data instances into equally sized groups, regardless of the actual structure of the data. In this case, setting $\beta = 0.5$ enabled the identification of meaningful clustering configurations, which were further validated through manual inspection and high internal validity scores.

3.5 Alternative Graph-Based Approach

To offer an alternative perspective on the groupings that may emerge in the data, a conceptually different approach was implemented. Rather than clustering attendees based on a dissimilarity metric derived from their aggregated attendance measures and spatio-temporal patterns, the problem was reframed as a bipartite graph for each night. Here, nodes are divided into two sets: *events* and *attendees*, with weighted edges representing the implicit scores attendees assigned to events through their physical presence, as detailed in Sec. 3.2.2.

From this framework, communities were detected by maximizing Barber's modularity for bipartite graphs (Eq.2.10) using the Louvain algorithm implementation from scikit-network [43]. Community detection was performed across multiple resolution values, ranging from 0.3 to 1.3 in increments of 0.1. The partitioning with the highest modularity was then compared to the dissimilarity-based clustering results for each night, analyzing the differences between the outcomes. This process is detailed in Sec.4.3.

One aspect to note about the network-driven strategy is that it is motivated by the methodology used in the previous study with Sónar's data [4]. However, a key difference is that, in that case, the authors constructed a graph where nodes represented *only artists*, as their goal was to segment them according to musical styles. In contrast, this approach remains centered on grouping attendees based on their event attendance patterns and the implicit connections they form through shared experiences.

This graph-based method relies on the same scores computed for the dissimilarity-based clustering, which are indicative of implicit feedback on music preferences. However, the $D_{combined}$ strategy also incorporates additional aspects of attendees' presence at the festival. Comparing these techniques can help evaluate whether music preference and interconnectivity with other entities at the festival account for attendees' behavior in other respects.

To formally assess the similarity between the resulting clusters or communities, three metrics were used. These metrics enable comparison without requiring ground truth labels and account for differences in the number of detected groups [44]:

- **Adjusted Rand Index (ARI):** Measures the similarity between two clusterings by considering all pairs of samples and counting those that are assigned consistently in both clusterings. The ARI is adjusted for chance, and is bounded above by 1 (perfect agreement), while 0 indicates a random assignments. It can yield negative values for particularly discordant configurations.
- **Normalized Mutual Information (NMI):** The relationship between two clusterings is quantified by measuring shared information, which refers to the extent to which knowing one clustering assignment reduces uncertainty about the assignment in the other. NMI is normalized to range from 0 (no mutual information) to 1 (perfect correlation).
- **Adjusted Mutual Information (AMI):** Like NMI, it is based on mutual information but accounts for similarity expected by chance. While NMI takes values in $[0, 1]$, AMI assigns 0 to random agreement, is bounded above by 1, and can have negative values.

4

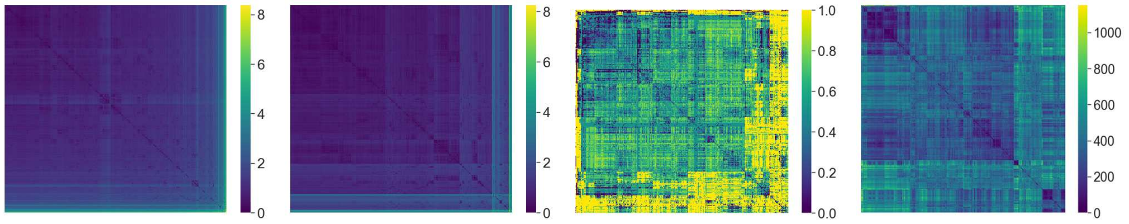
Results and Discussion

As the proposed formulation relies on combining variants of a time series dissimilarity measure, D_{TS}, with a distance derived from general features, it is insightful to evaluate the impact of different D_{TS} measures on cluster formation. This chapter first examines this aspect in Sec. 4.1, then presents and analyzes the clusters for both nights in Sec. 4.2, and finally compares the clustering results with the community detection approach on graphs in Sec. 4.3.

4.1 Impact of D_{TS}

4.1.1 Isolated behavior of each D_{TS}

Before analyzing the effect of each D_{TS} variant (CTWE, LCSS, and DTW) on $D_{combined}$ and its embedded representation, it is useful to first analyze how these dissimilarity measures behave independently and prior to projection. With this in mind, the distance matrices were reorganized hierarchically using agglomerative clustering [45], which rearranged the data according to pairwise similarities to obtain preliminary visualizations (Fig. 4.1). This method restructures the distance matrix to group closely related instances together, sometimes revealing distinct blocks or regions in the matrix (represented as a heatmap) that visually indicate potential clusters in the data. While agglomerative clustering can also serve as a clustering tool, it was employed here solely for preliminary visualization of the distance matrices. The actual clustering process is detailed in Sec. 3.3.4.



(a) CTWE matrix. **(b)** CTWE (PCA) matrix. **(c)** LCSS matrix. **(d)** DTW matrix.

Figure 4.1: Distance matrices of each D_{TS} on its own for night 2, reorganized hierarchically. Values shown are prior to normalization.

The D_{TS} matrices shown in Fig. 4.1 already offer an intuition of how these measures capture the similarities between trajectories in this context. First, CTWE produces a somewhat homogeneous distribution, without clear contrasting blocks, suggesting that it fails to find segmented groups. Moreover, most of the values are on the lower part of the range, which means that many instances are considered similar. This suggests that aggregating the time spent at each location into time windows offers a perspective that may be too coarse to differentiate between trajectories. Additionally, the contemporaneous restriction enforced by CTWE appears to have limited impact in this setting. Also, while the PCA reduction seems to do a better job at grouping small blocks, its benefit appears to be minimal.

In comparison, LCSS outputs highly contrasting values, but fails to uncover structured blocks. Its point-wise matching mechanism results in trajectories being either very close or very dissimilar. Given the small set of alternative events (there are only four stages), the dissimilarity metric based on aggregated binary matches yields a non-smooth distribution, and the dataset size and characteristics do not seem to provide enough information to reveal organized groupings.

On the other hand, the DTW matrix exhibits a more balanced distribution of distances across the range. This appears to blend contrast and smoothness, allowing the formation of blocks of varying sizes, characterized by smaller intra-block distances and larger inter-block distances. Although this visual inspection only provides a qualitative intuition about the effect of each D_{TS}, it suggests that DTW is a stronger candidate for identifying structured clusters based on the time-series aspect of trajectories.

While Fig. 4.1 displays the matrices for night 2 for brevity, the overall tendencies of all D_{TS} types remain consistent for night 1.

4.1.2 Effect of each D_{TS} on the overall $D_{combined}$

As described in Sec. 3.3, the variable λ regulates the influence of the time series dissimilarity measures, D_{TS}, on the overall measure of dissimilarity between trajectories. To individually evaluate the effect of each D_{TS} variant on the overall dissimlarity, Figures 4.2, 4.3, and 4.4 show the UMAP projections of $D_{combined}$ for the different λ values considered for night 2.

There are some important remarks regarding the UMAP projections before analyzing them. First, these figures refer to night 2 only because the effect of D_{TS} was slightly more evident; however, the overall phenomenon was similar for night 1. Second, for the $D_{combined}$ that involves CTWE, only the UMAP projection with the PCA version of CTWE is shown in Fig. 4.2 since the version that omitted the PCA reduction was quite visually similar. This suggests that PCA does not significantly impact this formulation. Lastly, all of these figures correspond to the UMAP parameters (`n_neighbors` and `min_dist`) fixed for clustering, as described in Sec. 3.3.5.

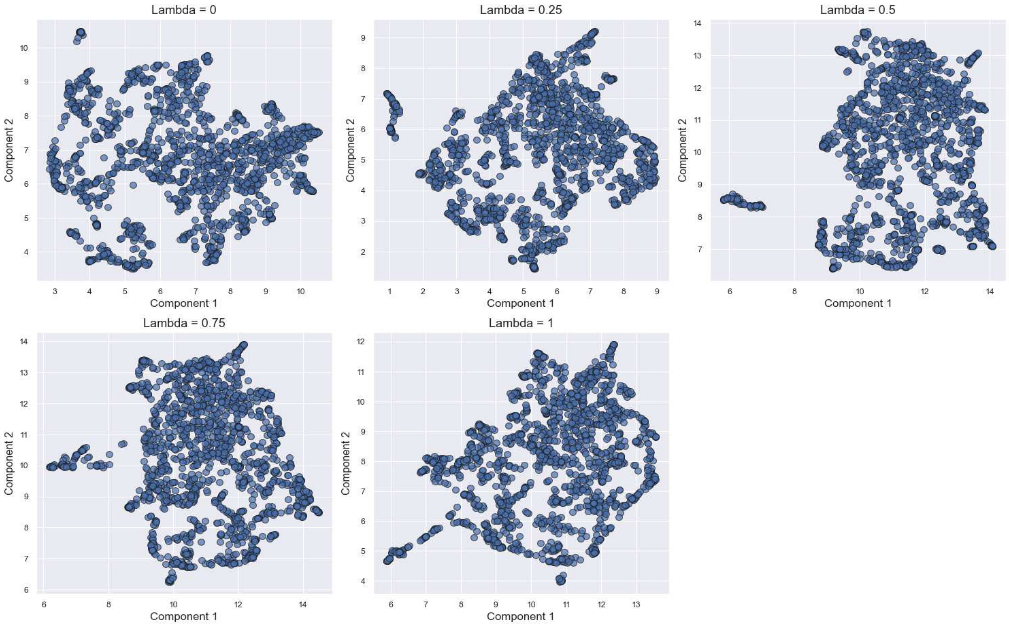


Figure 4.2: UMAP projections for different λ values. Combination of CTWE (with PCA reduction) and D_{GF} for night 2.

The visual inspection of the UMAP projections provides valuable insights into the differences between CTWE, LCSS, and DTW in terms of the configuration of the resulting embedded space. One initial observation is that all these dissimilarity measures tend to separate a small group of points from the rest of the data. This effect becomes evident only when D_{TS} starts

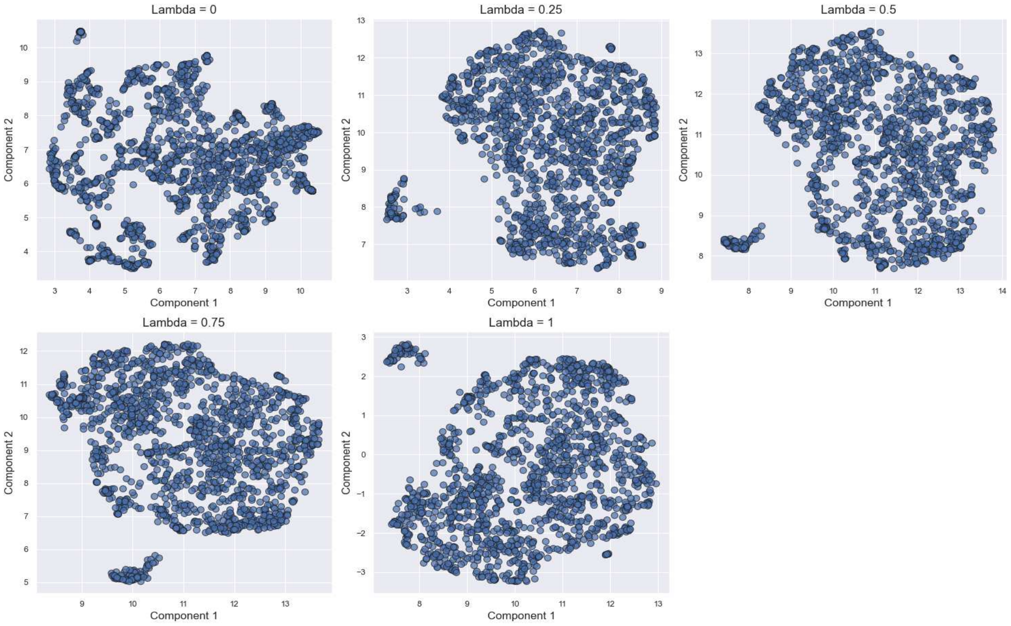


Figure 4.3: UMAP projections for different λ values. Combination of LCSS and D_{GF} for night 2.

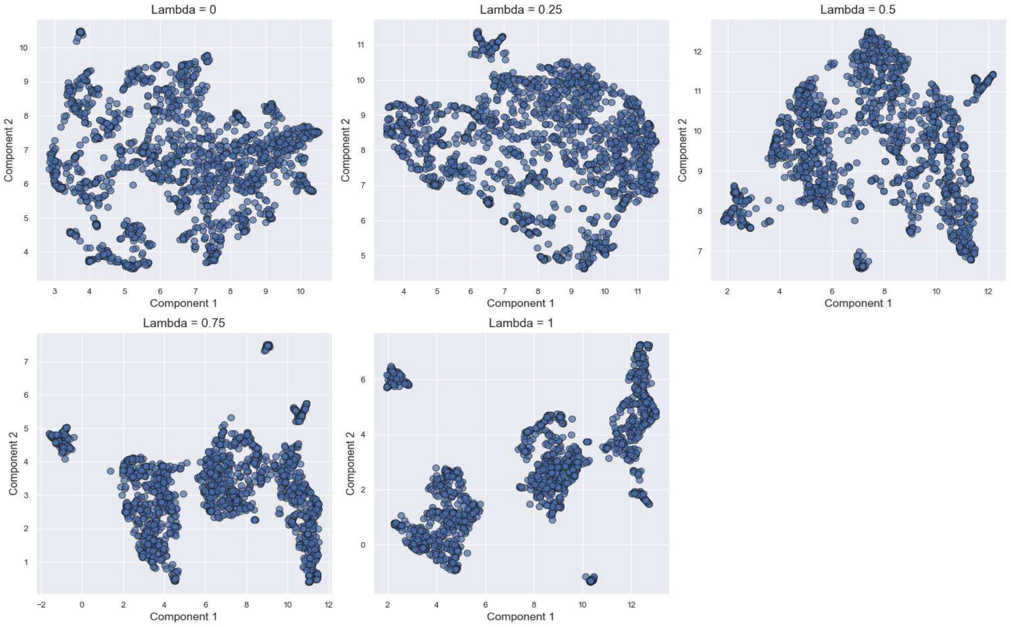


Figure 4.4: UMAP projections for different λ values. Combination of DTW and D_{GF} for night 2.

to influence the measure (i.e., when $\lambda > 0$) and persists across different values of λ.

Nevertheless, another important observation is that CTWE and LCSS appear to struggle in segmenting the data into well-separated clusters. Although CTWE seems to exhibit some structure in certain cases, neither of these distances provides clear insight into the number of clusters present in the data or the optimal λ value that might help identify them. Moreover, for most λ values, the $D_{combined}$ involving LCSS tends to form a homogeneous mass with no clear separation, apart from the small cluster mentioned above.

In clear contrast, Fig. 4.4 demonstrates that DTW imposes a distinct structure, which becomes more pronounced at higher λ values. This suggests that DTW not only aids in the formation of clusters but makes them more compact as λ increases (i.e., as DTW's influence strengthens). At $\lambda = 1$, at least three or four clusters are discernible, appearing, at least visually, to be better separated compared to other configurations.

This visual assessment positions the formulation involving DTW as a strong candidate for providing greater benefits to clustering algorithms compared to other D_{TS} alternatives. However, more detailed conclusions about the nature of these clusters cannot be drawn solely from visual inspection, as the coordinates within the UMAP embedding space lack a specific meaning [40]. The following sections explore the clustering performance and provide a detailed characterization of the resulting clusters. Since Night 2 had a slightly larger dataset, its results are presented first.

4.2 Clustering results

4.2.1 Clusters for Night 2

A summary of the quantitative results obtained from the clustering process for night 2 is shown in Tab. 4.1. From these results, there is a very important remark to make. Although many clustering configurations yielded DBCV values greater than 0.85 for the best scenario, this does not necessarily translate into valuable clustering results.

As outlined in the previous section, for many dissimilarity measures, the embedded space consisted of a homogeneous mass separated from a small group. This pattern was effectively identified by the HDBSCAN instances. While this result aligns with the visual inspection of the UMAP projections for the measures involving CTWE and LCSS, its practical utility remains limited. Fig. 4.5a illustrates this phenomenon with a clustering configuration from one LCSS combined measure projection. Although it corresponds to the highest DBCV value for

	HDBSCAN			K-medoids		
	Best λ	Best DBCV	Number of clusters	Best λ	Best Silhouette	Number of clusters
DTW	1	0.89	7	1	0.65	3
CTWE	0.5	0.88	2	0.25	0.39	3
CTWE (PCA)	1	0.86	2	0.5	0.40	2
LCSS	0.25	0.92	2	0	0.38	4

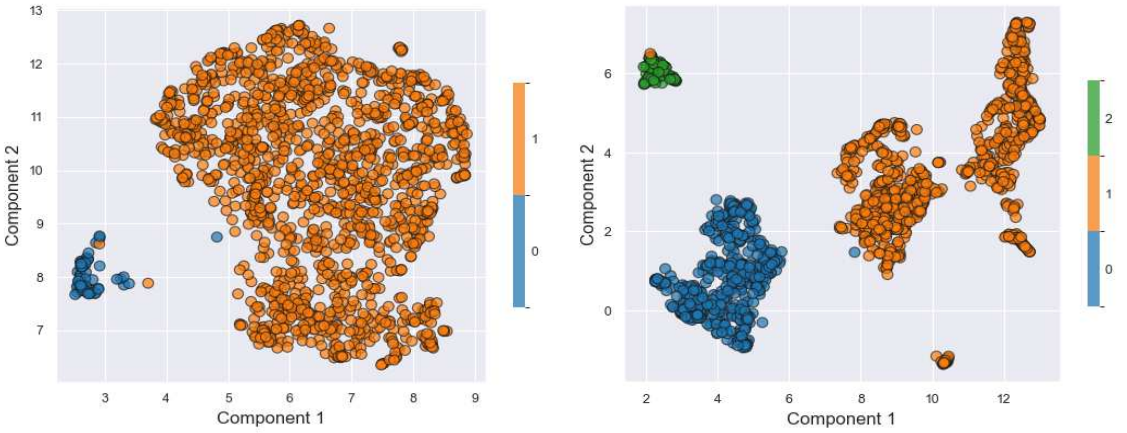
Table 4.1: Clustering results for night 2. The number of clusters exclude the noise category for HDBSCAN.

night 2, this configuration highlights the limitations of relying solely on validation metrics, as it provides minimal insights into the nuances of the underlying data. This is the general case for most of the results that yielded two clusters.

The issue discussed above arises from a dissimilarity measure that fails to adequately express the variability in the data in a way that enables the identification of distinct groups. Another suboptimal scenario occurs when the dissimilarity metric and its representation successfully segment the data structure, but the clustering algorithm is unable to fully capture these clusters. This is illustrated in Fig. 4.5b, which shows the UMAP projection for the measure involving DTW with a clustering assignment that fails to distinguish between two groups that visually appear distinct. This case, associated with the highest Silhouette score, demonstrates that K-medoids effectively identified the most suitable data representation for clustering, achieving a margin of at least 0.25 in the Silhouette score compared to alternative dissimilarity measures. However, it was unable to fully extract the clusters within this representation.

In contrast, a better clustering result in terms of internal validation metrics and cluster balance was obtained with HDBSCAN on the embedded space formed by the dissimilarity measure involving DTW, which is shown in Fig. 4.6a. This result supports the intuitive assessment described in Sec. 4.1, which identified DTW as the strongest candidate for uncovering meaningful structure in the data. Moreover, the clustering results indicate that increasing the influence of DTW enhances the data's clustering potential, as the highest λ value consistently produced superior internal validation metrics, in contrast with the other types of D_{TS}.

Although this result aligns well with visual expectations, it identifies some very small clusters that are challenging to analyze. Furthermore, other suitable clustering assignments that fit the data well and could better explain the underlying phenomena may remain undiscovered due to reliance on automatic evaluation with validation metrics and intensive hyperparameter search. Consequently, the BCS was applied to identify a more balanced clustering among the alterna-



(a) HDBSCAN with LCSS measure.

(b) K-medoids with DTW measure.

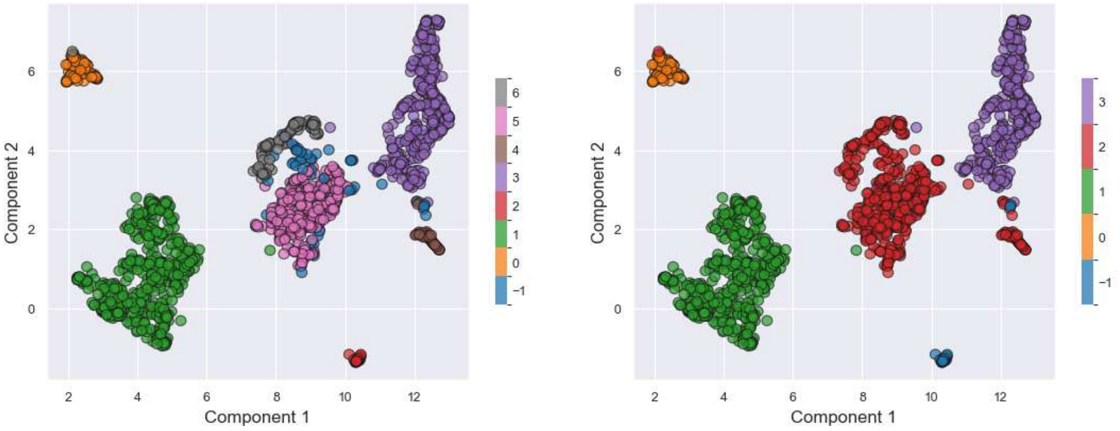
Figure 4.5: Suboptimal clustering configurations with high validation metrics for night 2. The figure on the left illustrates a poorly segmented data projection, effectively identified by HDBSCAN, but with limited practical value. The figure on the right depicts a well-segmented data representation that K-medoids failed to fully capture.

tives involving DTW, which proved to provide the most suitable structure. The outcome of this approach, shown in Fig. 4.6b, resulted in a BCS of 0.91 (compared to 0.89 for the configuration discussed above), while maintaining a high DBCV of 0.80. **Most importantly, this configuration provided more interpretable clusters for subsequent analysis.**

Once the best cluster configuration was selected to analyze each resulting group (the one in Fig. 4.6b), it became feasible to investigate the distinguishing features of the clusters. To this end, Tab. 4.2 summarizes key statistics that provide an initial understanding of some relevant measures. Also, to facilitate comparison across features with different units and scales, Fig. 4.7 shows a parallel coordinates plot with the means of the z-normalized features for each cluster.

Several patterns are immediately apparent from Tab. 4.2 and Fig. 4.7. First, clusters 1 and 2 spent more time, on average, at the festival. However, it is difficult to identify a clear trend in arrival or departure times based solely on the summary statistics, except for cluster 3, which arrived earlier than the other groups. Additionally, clusters 1 and 2 exhibit a higher number of attended events, greater straight-line distance traveled, and a larger radius of gyration. This suggests a more *exploratory* behavior for clusters 1 and 2 compared to clusters 0 and 3. Differences in other variables like average cell density and time spent in restaurant or bar zones are less consistent between groups and lack the expressiveness needed to characterize the clusters, but are included for completeness.

Fig. 4.8 provides a more comprehensive view on the time spent at the festival, as well as the



(a) Best clustering according to DBCV

(b) Best clustering according to BCS

Figure 4.6: Best clustering assignments for night 2. Both cases use the $D_{combined}$ involving DTW with $\lambda = 1$ and applied HDBSCAN for clustering. This is a 2D projection of clustering performed on 5D embeddings, which explains the few points appearing outside their clusters. Noise is labeled as -1.

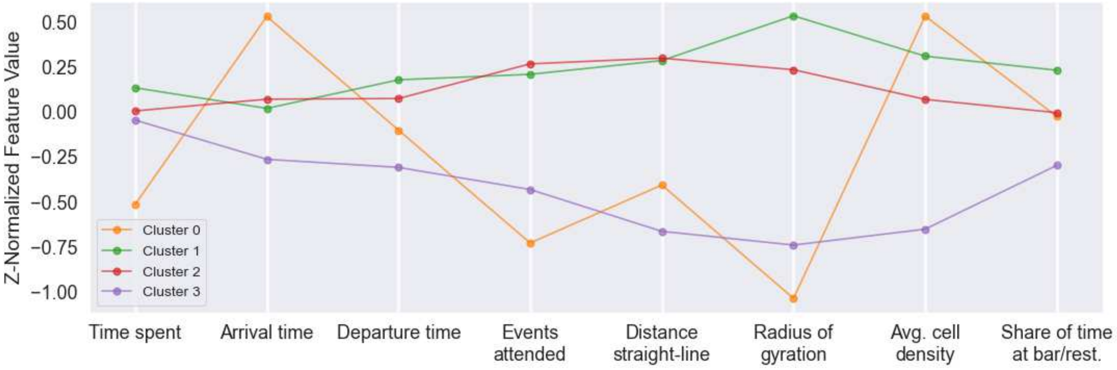


Figure 4.7: Parallel coordinates plot of cluster means (night 2). Each feature was z-normalized to make scales comparable.

arrival and departure times. The plot highlights the difference in the distribution of arrival times, making it clear that cluster 3 consists of attendees who typically arrive earlier than the other groups. Clusters 1 and 2 appear quite similar in this regard, while cluster 0 is too dispersed to reveal a distinct pattern. Meanwhile, differences in departure times are less evident, as all distributions peak around the same time—a little after the festival ends (610 minutes from the start). This indicates that most attendees stay until the end of the last event, with some leaving earlier, particularly around the 400- and 500-minute marks. However, no clear separation by cluster is observed in the departure times.

That said, one of the most salient aspects of these clusters concerns the movement patterns of attendees, which reveals one of the main conclusions of this thesis. To further explore this mat-

	Cluster 0 (Count=115)		Cluster 1 (Count=564)		Cluster 2 (Count=553)		Cluster 3 (Count=421)	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Time spent (min)	297.7	169.3	395.0	132.9	375.7	144.7	368.0	166.2
Arrival time (min)	260.6	161.2	198.6	97.3	204.8	118.6	164.2	132.4
Departure time (min)	558.3	148.3	593.5	104.3	580.4	122.3	532.2	141.5
Num. attended events	**2.70**	**1.53**	**4.90**	**2.31**	**5.00**	**2.30**	**3.38**	**1.91**
Dist. straight line (km)	**2.60**	**2.08**	**4.64**	**3.00**	**4.68**	**2.95**	**1.83**	**1.90**
Radius gyration (km)	**0.05**	**0.03**	**0.10**	**0.02**	**0.10**	**0.03**	**0.06**	**0.03**
Avg. cell density	7.31	3.35	6.74	2.53	6.12	2.23	4.27	1.92
Share of time at bar/rest.	0.25	0.27	0.31	0.24	0.25	0.20	0.18	0.25

Table 4.2: Cluster general statistics (means and standard deviations) for night 2. The arrival and departure times are measured with respect to the start of the festival.

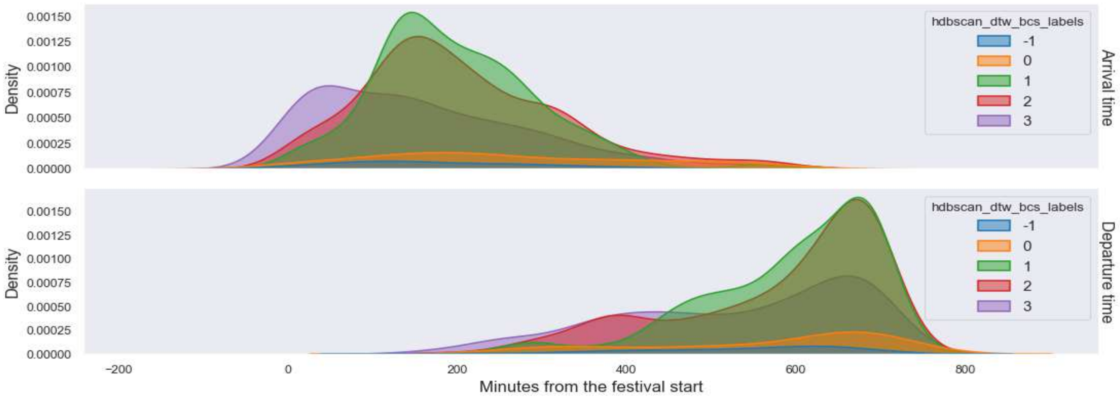


Figure 4.8: Arrival and departure times for each cluster (night 2), relative to the festival start. Noise is labeled as -1.

ter, Fig. 4.9 depicts the distributions of the radius of gyration by cluster. These distributions make it clearer that clusters 1 and 2 exhibit a tendency to move over a wider area than clusters 0 and 3. Also, a broad perspective of the relationship between the straight-line distance and the time spent at the festival is shown in Fig. 4.10. This plot shows that, even for a comparable range of durations, clusters 1 and 2 show a more pronounced tendency for greater distances traveled as time spent increases, compared to the other groups. This suggests that attendees in these clusters are more mobile and explore a wider range of activities and locations within the festival, which is consistent with a higher number of attended events.

At this point, the distinction between clusters 1 and 2, and clusters 0 and 3 is clear: the former pair of clusters can be characterized *explorers*, displaying greater movement, while the latter

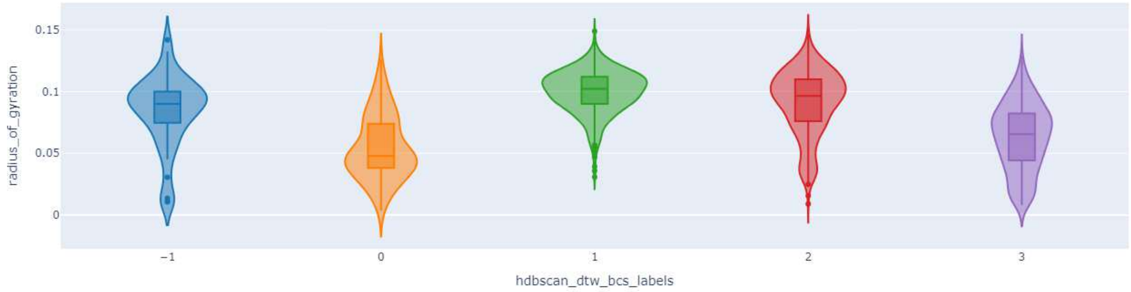


Figure 4.9: Radius of gyration for night 2 (in km). Distributions by cluster. Noise is labeled as -1.

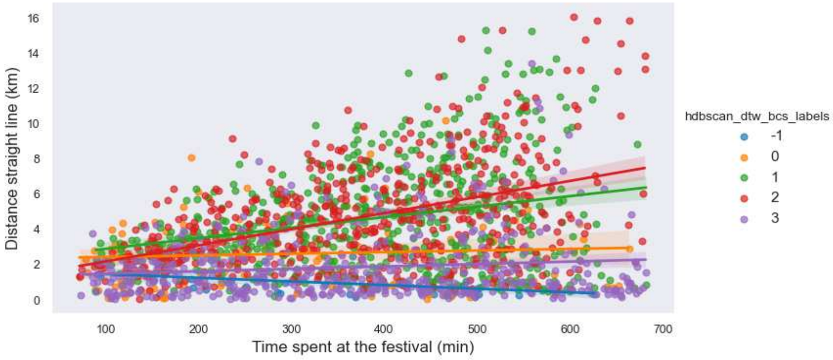
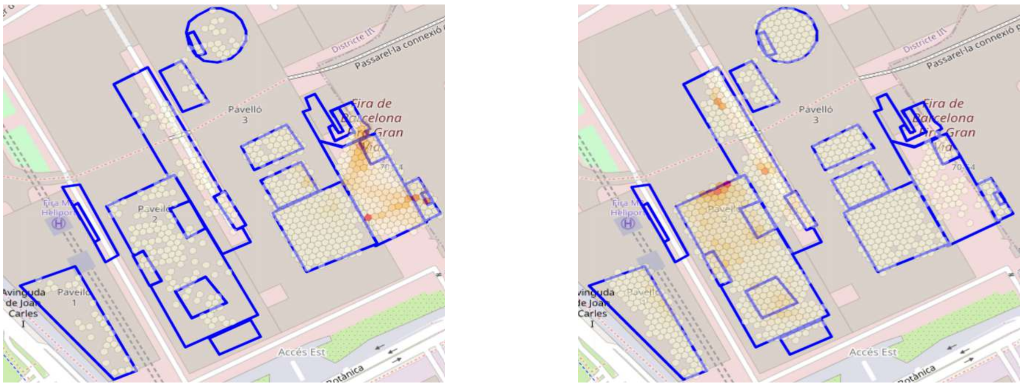


Figure 4.10: Straight-line distance versus time spent at the festival (night 2). Distributions by cluster. Noise is labeled as -1.

pair exhibits more restricted mobility. Nonetheless. further inspection can be performed to understand the differences between clusters 0 and 3. Aside from their sizes, both groups share a predominantly *stationary* nature. In this regard, Fig. 4.11, which shows the total observations per H3 cell for clusters 0 and 3, is useful for understanding *where* the attendees moved during the festival. Cluster 0 attendees moved predominantly within SonarPub and surrounding areas. In contrast, cluster 3 members mostly occupied the opposite parts of the venue, particularly the SonarClub.

The tendency of clusters 0 and 3 to remain primarily within SonarPub and SonarClub, respectively, is also evident over time in Figures 4.12 and 4.13. This finding highlights what distinguishes these two clusters, beyond the observation that cluster 3 is associated with early arrivals.

On the other hand, the phenomenon observed in Fig. 4.11 was not present among the *explorers* (clusters 1 and 2). Due to their higher mobility, these attendees moved across numerous locations during the night, making a static snapshot of their observation counts insufficient for understanding their behavior. Instead, inspecting their movement patterns over time reveals the characteristics that set them apart.



(a) Cluster 0 mostly moved within SonarPub (right stage) **(b)** Cluster 3 mostly moved within SonarClub (left stage)

Figure 4.11: H3 cell observation counts for clusters 0 and 3. Red indicates higher observation counts. The counts consider the full trajectories before condensing them into sequences of stops.

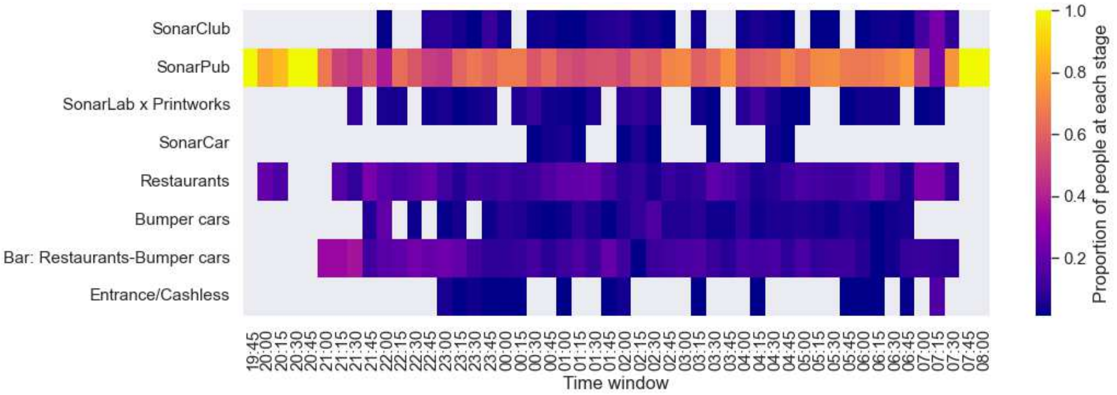


Figure 4.12: Proportion of unique MAC addresses relative to the total for each time window (cluster 0, night 2). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

Figures 4.14 and 4.15 illustrate the temporal distribution of attendees across different locations for clusters 1 and 2, respectively. While cluster 2 displays a more "chaotic" movement, with no evident homogeneous patterns, cluster 1 distinctly gravitates from SonarClub to SonarPub at a specific time (around 02:30), coinciding with performances by artists such as Floating Points, The Martinez Brothers and Kerri Chandler.

These findings provide insight into the effect of DTW in measuring the dissimilarity between trajectories. For clusters 0 and 3, DTW aids in segmenting trajectories that were confined within distinct spaces by leveraging the spatial aspect of its computation. For cluster 1, this measure (supported by the context provided by D_{GF}) appears to effectively capture the similarity among trajectories that follow a coherent route, with only slight back-and-forth vari-

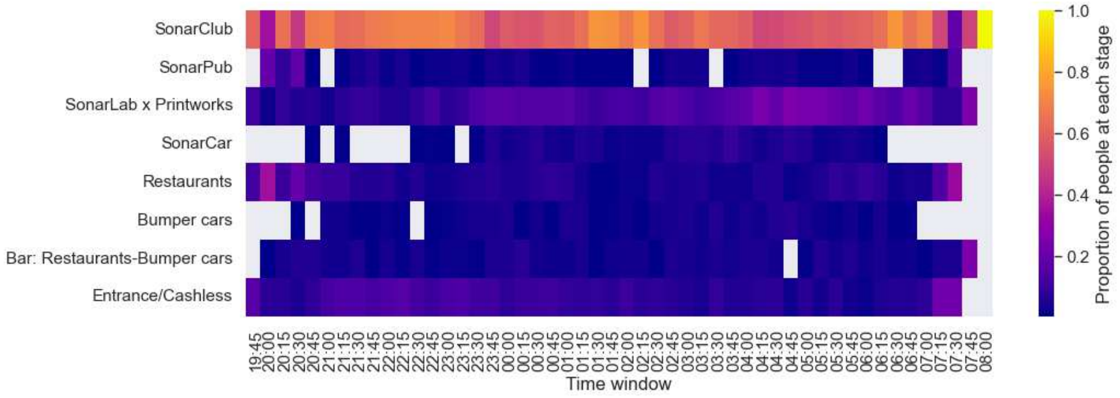


Figure 4.13: Proportion of unique MAC addresses relative to the total for each time window (cluster 3, night 2). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

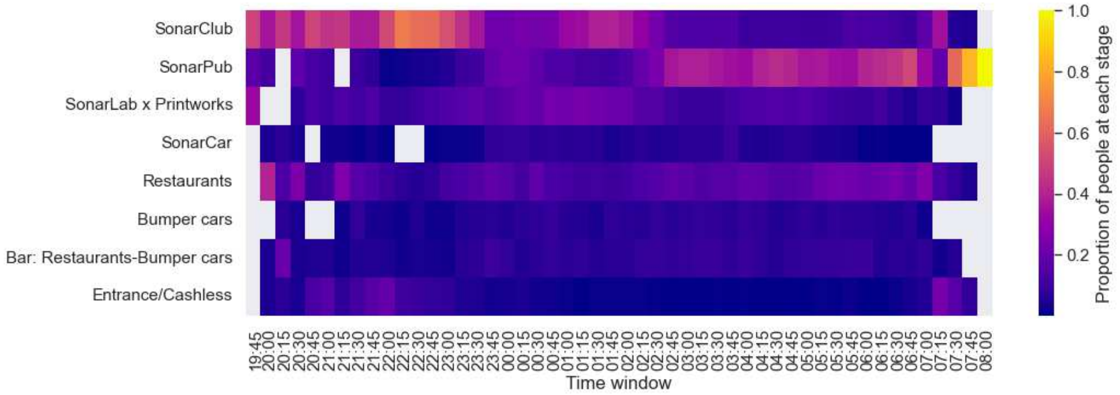


Figure 4.14: Proportion of unique MAC addresses relative to the total for each time window (cluster 1, night 2). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

ations. However, for more irregular and diverse movements, such as those in cluster 2, DTW struggles to distinguish between trajectories that may have similar overall shapes but do not progress simultaneously. This effect is likely due to its alignment mechanism. Nonetheless, DTW clearly outperformed other D_{TS} measures in identifying structure within the data and contributed to shape interpretable and insightful clusters.

On the other hand, one aspect yet to be discussed is attendees' *musical preference*, which is intended to be captured by incorporating the scores they implicitly assigned to each event, as described in Sec. 3.2.2). Since these scores influence the overall dissimilarity via D_{GF}, they affect cluster formation in ways that DTW alone does not capture. Evidence of this emerges when ranking events within each cluster by aggregating the adjusted scores, following the approach in [4]. As shown in Figures A.5 and A.6 of the appendix, clusters 1 and 2 (the *explorers*) exhibit

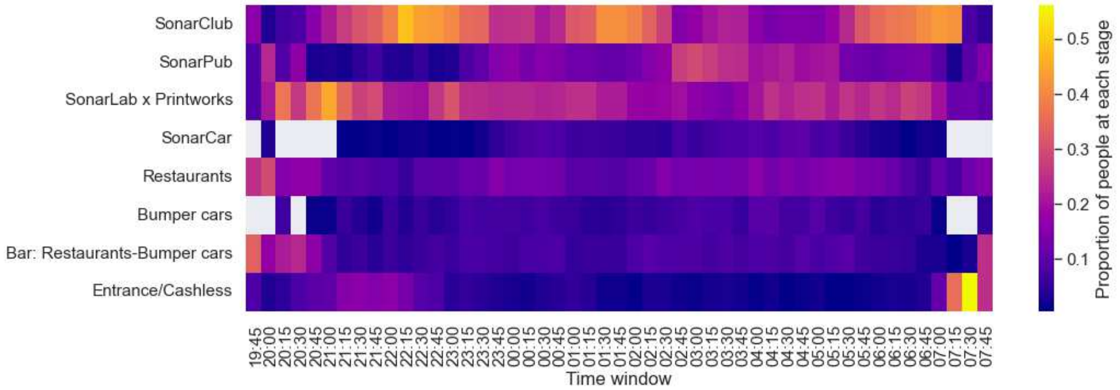


Figure 4.15: Proportion of unique MAC addresses relative to the total for each time window (cluster 2, night 2). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

distinct event preferences, further differentiating them. For the two stationary clusters (0 and 3), which spent most of their time at a single stage, the rankings are naturally dominated by the artists who performed there.

4.2.2 Clusters for Night 1

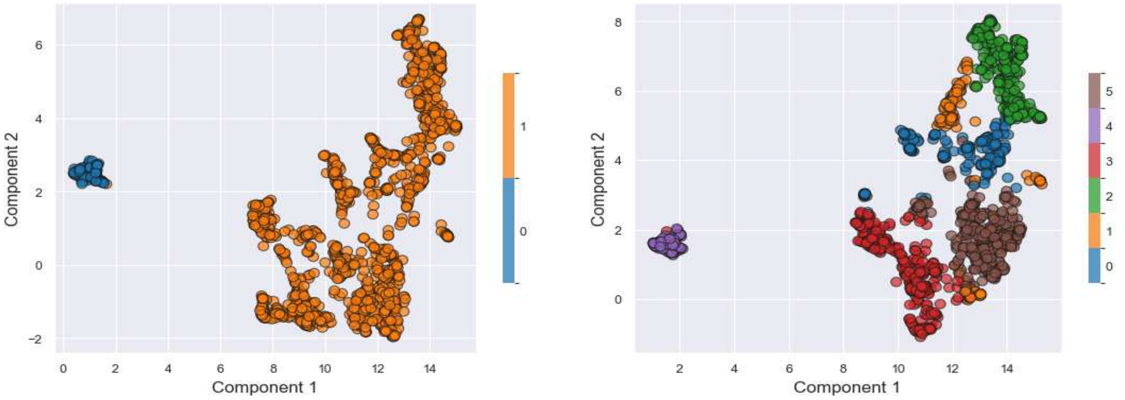
Although some of the outcomes of the dissimilarity-based clustering approach are more notable for night 2, some overall tendencies are also observed for night 1. However, there are some aspects that are specific for night 1.

First, as shown in Tab. 4.3, DTW was the best-performing D_{TS} based on both internal metrics: DBCV and Silhouette score. Moreover, the best DTW configurations corresponded to higher λ values compared to other time series dissimilarity measures, reinforcing the idea that DTW's influence enhances cluster formation. These results confirm previous findings and further establish DTW as the most suitable D_{TS} for identifying patterns in this context.

	HDBSCAN			K-medoids		
Type of D_{TS}	Best λ	Best DBCV	Number of clusters	Best λ	Best Silhouette	Number of clusters
DTW	1	0.95	2	0.75	0.49	6
CTWE	1	0.95	2	0.25	0.37	2
CTWE (PCA)	0.75	0.87	2	0.25	0.37	2
LCSS	0.25	0.65	2	0	0.37	2

Table 4.3: Clustering results for Night 1. The number of clusters exclude the noise category for HDBSCAN.

However, the best performing configurations found by HDBSCAN and K-medoids, shown in Fig. 4.16, suffer from one of two issues. Either only two clusters are identified, as in Fig. 4.16a, with one appearing to contain some internal substructure; or the data is fragmented into many small subgroups that are difficult to analyze due to their limited representativeness, as seen in Fig. 4.16b.



(a) Best HDBSCAN clustering according to DBCV

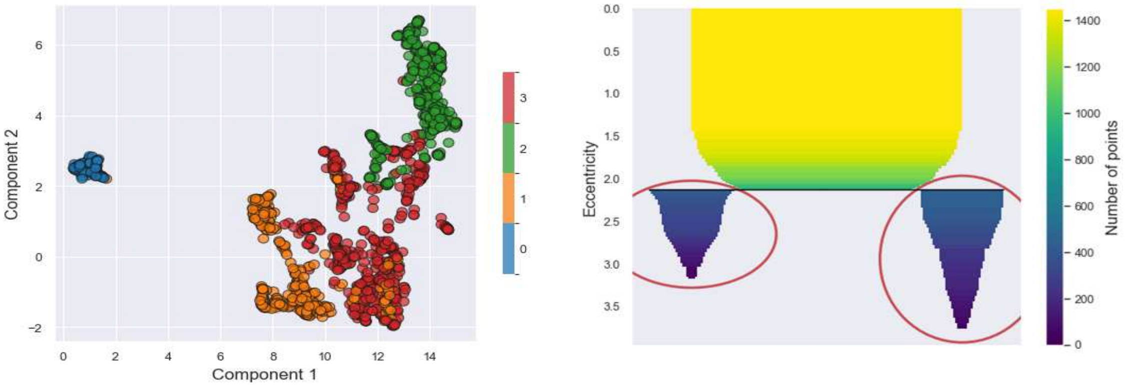
(b) Best K-medoids clustering according to Silhouette score

Figure 4.16: Best clustering assignments for night 1. Both cases use the $D_{combined}$ involving DTW, with the left configuration using $\lambda = 1$ and the right configuration using $\lambda = 0.75$.

For this night, using BCS to find a more balanced clustering also led to numerous small clusters that were unfeasible to analyze. However, one useful feature of HDBSCAN helped address this issue. As a hierarchical clustering technique, HDBSCAN identifies substructures within clusters. In this case, although the best clustering configuration found by HDBSCAN for night 1 only forms two clusters, it is visually apparent that one cluster contains subgroups which appear generally connected but show some degree of separation. Consequently, the hierarchical structure of HDBSCAN was leveraged to detect these sub-clusters, or *branches*.

In short, given an HDBSCAN instance, branch detection is a post-processing step which identifies regions where a cluster exhibits internal branching structure, segmenting it into stable subgroups. Instead of density, this process detects branches using *eccentricity*, a measure of the distance of a point from its cluster centroid. A hierarchy of in-cluster eccentricity thresholds is constructed, each delineating branch structures from the core, and then a stable branch membership configuration is selected. The branches detected in the best HDBSCAN clustering configuration for night 1 are shown in Fig. 4.17.

The groups resulting from branch detection, in Fig. 4.17a offer a more balanced configuration that was more feasible to analyze qualitatively than the previous alternatives. Tab. 4.4



(a) Detected Branches for best HDBSCAN clustering.

(b) Eccentricity contour plot for branches in cluster 1.

Figure 4.17: Branch detection in the best HDBSCAN clustering for night 1 according to DBCV. The right panel shows how the "core" branch splits into two sub-branches for cluster 1 of original configuration. The left panel is a 2D projection of clustering performed on 5D embeddings, which explains the few points appearing outside their cluster/branch.

provides a summary of statistics describing these groups (which will be referred to interchangeably as branches or clusters), while Fig. 4.18 shows the corresponding parallel coordinates plot.

	Cluster 0 (Count=128)		Cluster 1 (Count=455)		Cluster 2 (Count=441)		Cluster 3 (Count=556)	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Time spent (min)	355.6	162.7	359.8	128.6	366.1	177.3	356.2	123.7
Arrival time (min)	180.9	142.3	195.9	103.0	153.8	129.3	199.6	103.7
Departure time (min)	536.5	113.0	556.0	120.0	520.0	131.1	555.8	104.0
Num. attended events	**2.62**	**1.40**	**4.85**	**2.33**	**2.64**	**1.44**	**4.97**	**2.12**
Dist. straight line (km)	**2.53**	**2.12**	**4.74**	**2.95**	**1.25**	**1.23**	**4.53**	**2.57**
Radius gyration (km)	**0.05**	**0.03**	**0.10**	**0.02**	**0.07**	**0.03**	**0.09**	**0.02**
Avg. cell density	6.17	2.83	5.65	1.82	3.43	1.50	5.46	1.93
Share of time at bar/rest.	0.28	0.29	0.24	0.21	0.22	0.30	0.23	0.20

Table 4.4: Cluster general statistics (means and standard deviations) for night 1. The arrival and departure times are measured with respect to the start of the festival.

Interestingly, the selected clusters from night 1 and night 2 exhibit similar characteristics. In both cases, some attendee groups were more stationary (clusters 0 and 2 on night 1), while others were more exploratory (clusters 1 and 3 on night 1). For brevity, not all plots for night 1 are shown, as they closely resemble those from night 2. Specifically, the radius of gyration and straight-line distance depictions were nearly identical to Figures 4.9 and 4.10, differing only in

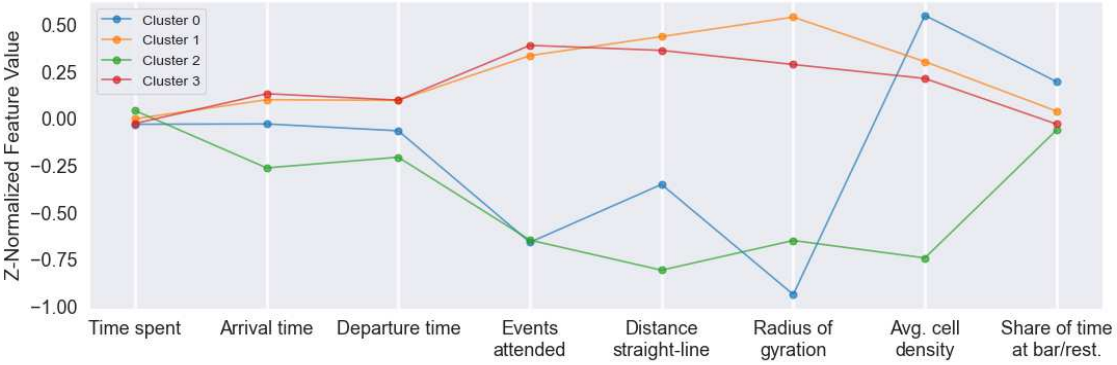


Figure 4.18: Parallel coordinates plot of cluster means (night 1). Each feature was z-normalized to make scales comparable.

the cluster labels.

This result suggests the existence of typical mobility patterns among Sónar by Night attendees, likely influenced by the venue layout and the structured lineup, which follows a similar format on both nights. Nonetheless, these patterns were more easily identified in night 2, whereas for night 1 it was necessary to further divide one of the initial clusters into separate branches. This difference could be due to the slightly larger dataset for night 2 (which may have improved pattern generalization) or intrinsic variations in attendee behavior between the two nights.

Aside from the common trends on the aggregated mobility measures and the characterization of explorers and stationary participants, it is also useful to examine what distinguishes each individual cluster from the others, similar to what was done for night 2. The distribution of arrival and departure times, shown in Fig. 4.19, reveals no clear distinguishing pattern for departure times. In contrast, one group (cluster 2 in this case) tends to arrive earlier than the others. This observation mirrors the findings for night 2, differing only in cluster labels.

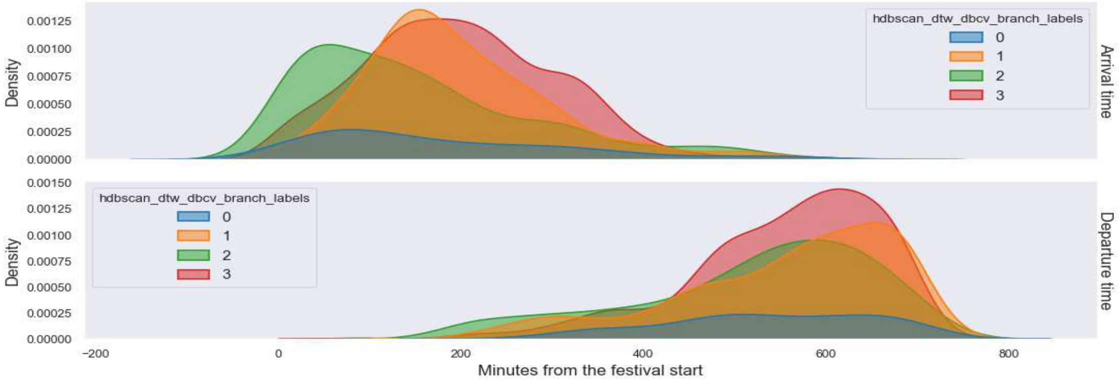


Figure 4.19: Arrival and departure times for each cluster (night 1), relative to the festival start.

The areas typically frequented by each cluster also exhibit parallels to night 2. As depicted in Figures 4.20 and 4.21, stationary attendees remained within one of the two main stage areas (SonarClub or SonarPub). This phenomenon further reinforces the idea that Sónar by Night attendees share some common mobility patterns across the two nights, which correspond to specific locations of the venue layout. However, each night has its particularities.

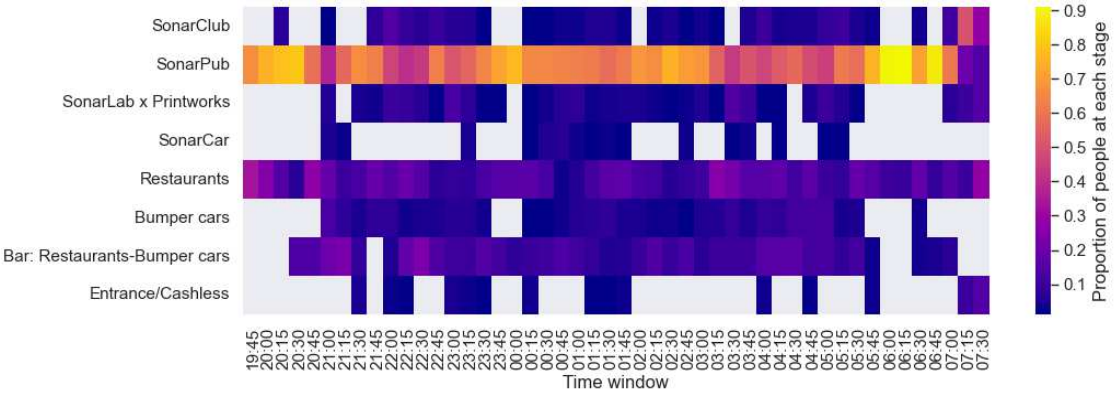


Figure 4.20: Proportion of unique MAC addresses relative to the total for each time window (cluster 0, night 1). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

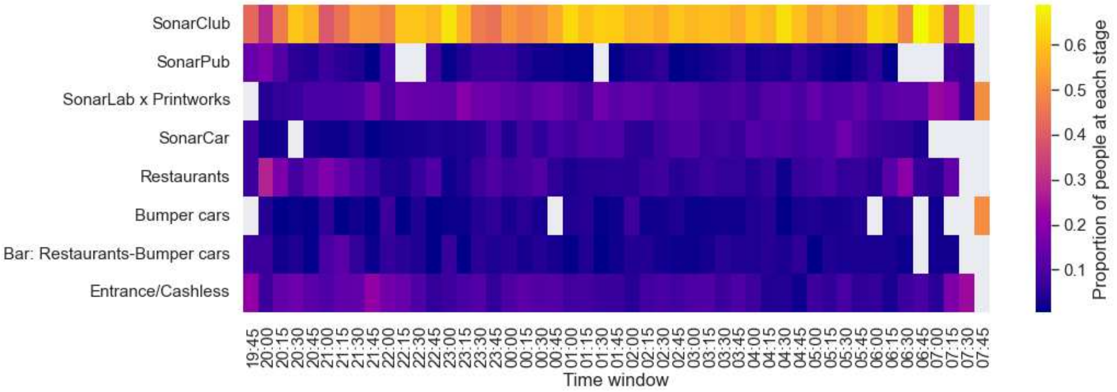


Figure 4.21: Proportion of unique MAC addresses relative to the total for each time window (cluster 2, night 1). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

Shaped by the distinct context of night 1, a group of explorers (cluster 1) appears to move together between SonarClub and SonarPub at specific moments, as shown in Fig. 4.22. These movements coincide with the acts of Jessie Ware, Kaytranada and Ben Böhmer, three of the evening's main acts. In comparison, cluster 3 (Fig. 4.23) exhibits more irregular movements that are difficult to characterize. While some alternation across stages is observed, it does not follow the same pendular pattern between SonarClub and SonarPub. Additionally, a significant portion of these attendees also visited SonarLab x Printworks, but without a clear temporal structure. Although these groups emerge from separate branches of the initial clustering—making distinctions between them less pronounced—this result suggests that, despite its ability to capture structured movement patterns, DTW struggles to differentiate trajectories with similar overall shapes that unfold at different times, leading to non-uniform aggregations.

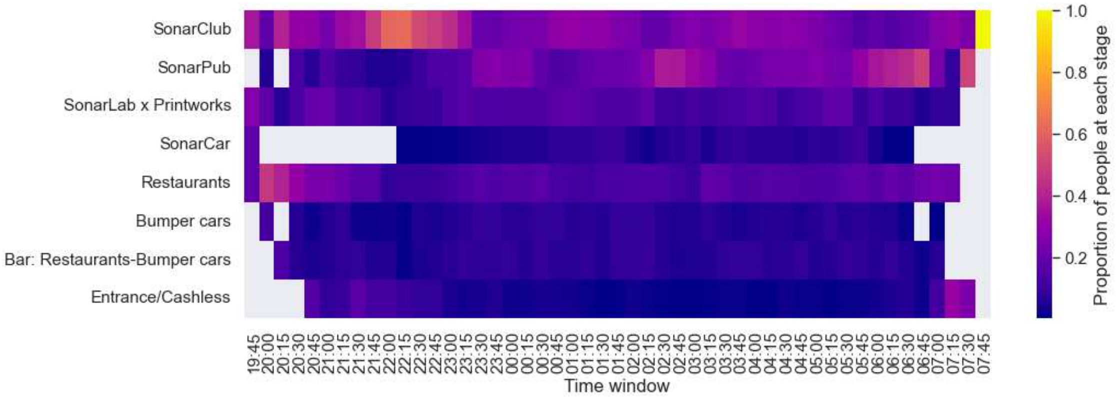


Figure 4.22: Proportion of unique MAC addresses relative to the total for each time window (cluster 1, night 1). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

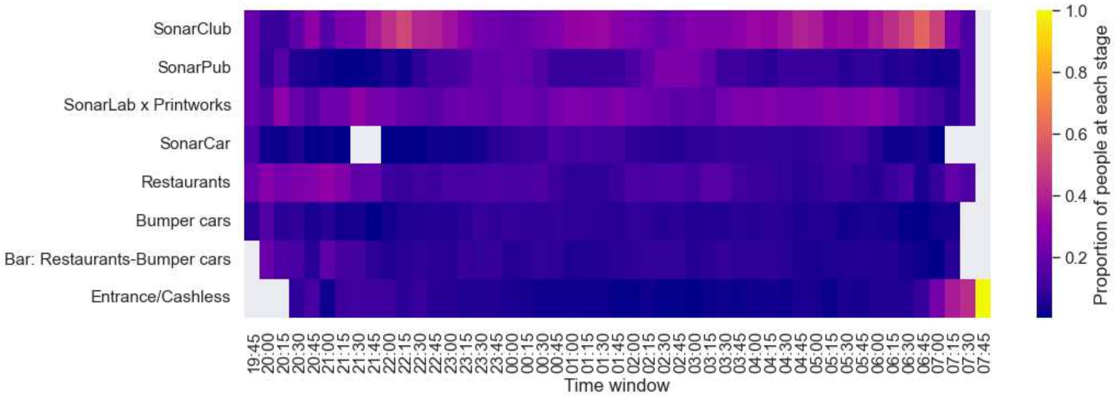


Figure 4.23: Proportion of unique MAC addresses relative to the total for each time window (cluster 3, night 1). Each column in the grid sums up to 1. The calculation includes only members of the corresponding cluster.

As with night 2, the event rankings that reflect musical preference are a distinguishing factor between clusters. Since the rankings for the stationary participants (clusters 0 and 2 for night 1) are dominated by the artists in their respective prevalent stages, only the rankings for clusters 1 and 3 are presented in Figures A.7 and A.7 of the appendix, respectively.

The findings in this section demonstrate that the dissimilarity-based clustering approach effectively produced meaningful and interpretable clusters. This was achieved by leveraging the capability of DTW to capture spatiotemporal similarities (contextualized by D_{GF}) and the robustness of HDBSCAN in detecting clusters with varying shapes and hierarchical structures. On the other hand, Sec. 4.3 discusses a radically different approach to find groups of attendees based on the graph representation described in Sec. 3.5.

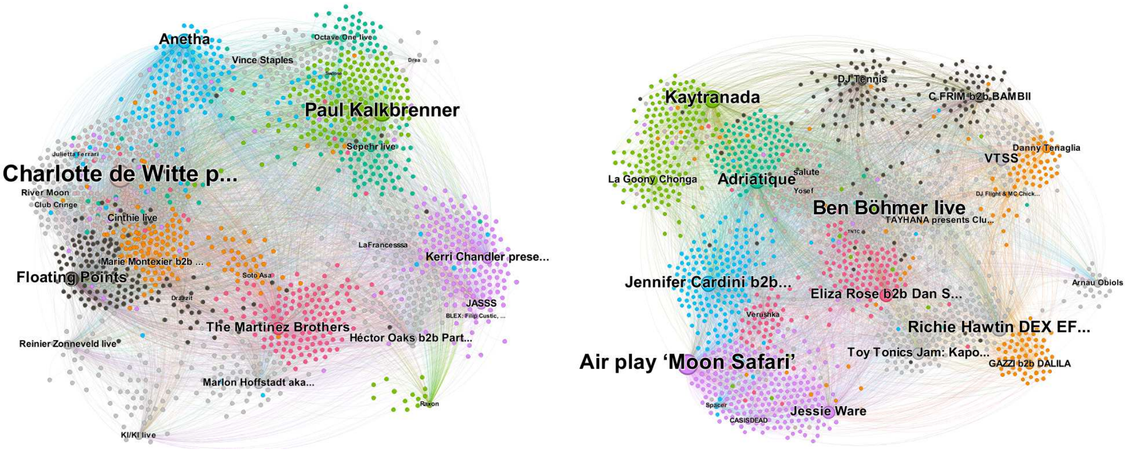
4.3 Comparison With Community Detection

Regarding the alternative graph-driven approach, the results of maximizing modularity for the best-performing resolutions are summarized in Tab. 4.5. Notably, the number of communities detected at optimal modularity values exceeds the number of clusters identified using the $D_{combined}$ formulation and clustering algorithms in the projected space. Additionally, lower modularity values are associated with fewer detected communities.

Night 2			Night 1		
Resolution	Number of communities	Modularity	Resolution	Number of communities	Modularity
1.1	16	0.4930	1.2	14	0.5218
1.0	16	0.4928	1.0	13	0.5217
1.2	18	0.4918	1.1	15	0.5215
0.9	13	0.4917	0.9	13	0.5214
1.3	19	0.4904	0.8	11	0.5196
0.8	10	0.4844	1.3	17	0.5176
0.7	8	0.4759	0.7	10	0.5156
0.6	6	0.4544	0.6	7	0.4995
0.5	5	0.4081	0.5	6	0.4731
0.4	3	0.3345	0.4	4	0.3648

Table 4.5: Community detection results for two nights.

The best configurations yield modularity values around 0.5, indicating a significant community structure. In practice, modularity values between approximately 0.3 and 0.7 are associated with strong community structures [25]. As the goal here is to compare the similarity of the resulting clusters/communities obtained through two very different approaches, the configuration selected for comparison is simply the one with the highest modularity for each night. This approach avoids initial induced bias and relies on a validation metric. In this context, the number of communities detected already reveals a difference between the two approaches. Visually, these selected communities can be interpreted as nodes of attendees connecting with the events and gravitating around them, as depicted in Fig. 4.24.



(a) Bipartite graph of night 2.

(b) Bipartite graph of night 1.

Figure 4.24: Communities detected in attendee-event bipartite graphs for nights 1 and 2. The size of the nodes corresponds to their weighted degrees, and labels are assigned only to the events. Different colors denote communities for each night.

Since the graph formulation is based solely on the scores implicitly assigned by attendees to events, the community detection strategy places a stronger emphasis on music preference. While segmenting artists by musical style is not straightforward given the festival's experimental nature and the dominance of electronic music, this approach appears to successfully uncover some structure that reveals similarities between artists and how attendees relate to events. For example, the group clustered around Air, Jessie Ware, and CASISDEAD (represented by purple nodes in the Night 1 graph) suggests a community with a preference for events that deviate slightly from "pure" electronic music, leaning toward artists that incorporate elements from genres like pop or hip-hop. However, given the smaller cluster sizes and the subjective nature of genre classification, the results are less interpretable than those obtained through the $D_{combined}$ clustering approach.

The results for each night of the cluster similarity metrics (described in Sec. 3.5), considering the selected clustering and community configurations, are shown in Tab. 4.6.

The low values in the clustering comparison metrics (relative to their respective scales) indicate that the two approaches for identifying related groups differ significantly. Since this is an unsupervised setting, there is no single objectively correct strategy. Instead, the results sug-

	Night 2	Night 1
ARI	0.032	0.038
NMI	0.116	0.109
AMI	0.108	0.103

Table 4.6: Metrics for comparing the clustering and community detection approaches.

gest that each method provides distinct insights. Both approaches were derived from the same dataset and even rely on scores calculated in an identical manner. However, the $D_{combined}$ approach incorporates general behavioral measures of festival attendees, as well as spatio-temporal nuances in their trajectories. This provides a broader perspective on attendees' profiles, incorporating, but not limited to, their event preferences. In contrast, the community detection approach on the graphs emphasizes musical preferences and how these implicitly connect different events and attendees.

As a quick test, the same comparison metrics were computed for community configurations with a number of communities closer to the number of clusters derived from the $D_{combined}$ approach. These metrics yielded even lower values, suggesting that the resulting structure was less related to the one obtained through the $D_{combined}$ formulation (which also incorporates event scores). Along with the lower modularity values, which reflect weaker community structures, these configurations were ultimately left out of the final analysis.

5 Conclusion

This study developed and documented a complete methodology for analyzing movement behaviors and extracting attendee profiles at a music festival using anonymized Wi-Fi traces. By designing a framework that integrates general features with regulated trajectory clustering techniques, including the definition of context-specific sequences of stops, it was possible to evaluate the effectiveness of different dissimilarity measures in forming distinct clusters.

Specifically, DTW proved to be an effective time-series dissimilarity measure in this setting, clearly outperforming alternatives such as LCSS and an adaptation of the Euclidean distance for contemporary time windows. The formulation involving DTW made it possible to obtain distinguishable and interpretable clusters, not only differentiating attendees by their arrival times and musical preferences but also highlighting distinct mobility behaviors. Attendees were observed to split between *explorers*, who moved across multiple venue areas, and more *stationary* participants, who tended to remain within one of the two main audience zones. Among the explorers, some followed regular movement patterns between venue sections, while others displayed more irregular trajectories.

Additionally, the proposed framework produced different grouping configurations and provided distinct insights compared to a community detection approach based on graphs, which leveraged implicit feedback from attendees' physical presence at events. While the resulting clusters captured musical preferences, they also accounted for broader behavioral patterns and movement dynamics. In contrast, the graph-based method was more strongly driven by musical affinity to form smaller groups.

However, this study has limitations that could be addressed in future work, enhancing its applicability to similar settings. For instance, one challenge is distinguishing irregular and highly variable movement patterns. Despite being the best-performing time-series dissimilarity measure considered, DTW's alignment mechanism can struggle to differentiate movements with similar shapes that occur at different times. Since this study demonstrates that the proposed methodology effectively identifies relevant patterns in a music festival setting, this issue could be mitigated by adapting its key principles to models with greater abstraction power, such as neural architectures, while carefully controlling design choices.

In any case, to improve generalization and experiment with the proposed strategies under different conditions, future studies could benefit from larger and more purposefully collected datasets for this type of analysis. Since noise isolation was particularly time-consuming and the sparseness of some trajectories may limit the data representativeness, exploring alternative datasets might reveal additional methodological nuances and create opportunities for investigating various modeling approaches. Future research could also explore how insights from trajectory clustering in this context contribute to other tasks, such as trajectory generation and crowd flow prediction, even in a potential real-time setting, which could be valuable for event management.

References

[1] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, May 2015. [Online]. Available: https://doi.org/10.1145/2743025

[2] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, "Human mobility: Models and applications," *Physics Reports*, vol. 734, p. 1–74, Mar. 2018. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2018.01.001

[3] M. Yue, Y. Li, H. Yang, R. Ahuja, Y.-Y. Chiang, and C. Shahabi, "Detect: Deep trajectory clustering for mobility-behavior analysis," 2020. [Online]. Available: https://arxiv.org/abs/2003.01351

[4] J. C. Carrasco-Jiménez, F. M. Cucchietti, A. Garcia-Saez, G. Marin, and L. Calvo, "We know what you did last sonar: Inferring preference in music from mobility data," in *Intelligent Computing*, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham: Springer International Publishing, 2019, pp. 43–61.

[5] S. Festival, "Sónar program schedule," 2024, accessed: 2025-03-31. [Online]. Available: https://web.archive.org/web/20240622213211/https://sonar.es/en/programme/schedule

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[7] M. Rossi. (2023) Human data analytics slides. Lecture notes from the course "Human Data Analytics". [Online]. Available: https://stem.elearning.unipd.it/enrol/index.php?id=6756

[8] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 166–171.

[9] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW).* IEEE, Nov. 2017, p. 33–42. [Online]. Available: http://dx.doi.org/10.1109/ICDMW.2017.12

[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[11] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[12] scikit-learn developers, *HDBSCAN Clustering*, 2024, documentation for the scikit-learn library, version 1.5. [Online]. Available: https://scikit-learn.org/1.5/modules/clustering.html#hdbscan

[13] M. Luca, G. Barlacchi, B. Lepri, and L. Pappalardo, "A survey on deep learning for human mobility," 2021. [Online]. Available: https://arxiv.org/abs/2012.02825

[14] A. Leick, L. Rapoport, and D. Tatarnikov, *Geodesy*. John Wiley & Sons, Ltd, 2015, ch. 4, pp. 129–206. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119018612.ch4

[15] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3880–3887.

[16] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2017. [Online]. Available: https://doi.org/10.1007/s10462-016-9477-7

[17] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132031000453X

[18] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice-Hall Signal Processing Series, 1993.

[19] S. Salvador and P. Chan, "Fastdtw: Toward accurate dynamic time warping in linear time and space," in *KDD workshop on mining temporal and sequential data*. Citeseer, 2004.

[20] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 312–319.

[21] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings 18th International Conference on Data Engineering*, 2002, pp. 673–684.

[22] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *J. Intell. Inf. Syst.*, vol. 27, pp. 267–289, 11 2006.

[23] R. Arthur, "Modularity and projection of bipartite networks," *Physica A: Statistical Mechanics and its Applications*, vol. 549, p. 124341, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437120301151

[24] J. Leskovec, "Lecture 14: Community structure in networks," https://snap.stanford.edu/class/cs224w-2021/slides/14-communities.pdf, 2021, accessed: 2025-03-25.

[25] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.026113

[26] M. J. Barber, "Modularity and community detection in bipartite networks," *Physical Review E*, vol. 76, no. 6, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.76.066102

[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1088/1742-5468/2008/10/P10008

[28] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of mac address randomization in mobile devices and when it fails," 2017. [Online]. Available: https://arxiv.org/abs/1703.02874

[29] V. Contributors, "Avoid getting duplicate mac address with mac address randomizer?" https://superuser.com/questions/1406516/avoid-getting-duplicate-mac-address-with-mac-address-randomizer, 2023, accessed: 2025-01-30.

[30] I. M. W. Group, "Randomized and Changing MAC Address," Internet Engineering Task Force (IETF), Tech. Rep. draft-ietf-madinas-mac-address-randomization-08, 2023. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-madinas-mac-address-randomization-08.html#name-os-current-practices

[31] I. Vasilevski, D. Blazhevski, V. Pachovski, and I. Stojmenovska, "Five Years Later: How Effective Is the MAC Randomization in Practice? The No-at-All Attack," in *Communications in Computer and Information Science*. Springer, 2019, ch. 5.

[32] IEEE Registration Authority, "Assignments," 2023. [Online]. Available: https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries

[33] J. Sabaté, "Sónar cierra la edición 2024 con récord de asistentes, hardcore, tecno y mucha danza," *elDiario.es*, 2024. [Online]. Available: https://www.eldiario.es/cultura/sonar-cierra-edicion-2024-record-asistentes-hardcore-tecno-danza-cat_129_11452303.html

[34] Purple.ai, "Mac randomization," https://support.purple.ai/hc/en-gb/articles/7330834299805-MAC-Randomization, 2023.

[35] A. O. S. Project. (2025) MAC Randomization Behavior. [Online]. Available: https://source.android.com/docs/core/connect/wifi-mac-randomization-behavior

[36] H. E. Staff, "How fast can a human run? plus, how to run faster," 2023, accessed: 2025-02-13. [Online]. Available: https://www.healthline.com/health/how-fast-can-a-human-run

[37] L. Pappalardo, F. Simini, G. Barlacchi, and R. Pellungrini, "Scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data," 2021. [Online]. Available: https://arxiv.org/abs/1907.07062

[38] L. McInnes, J. Healy, and J. Melville, "Using umap for clustering," 2024, accessed: 2025-03-15. [Online]. Available: https://umap-learn.readthedocs.io/en/latest/clustering.html

[39] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, *Density-Based Clustering Validation*, pp. 839–847. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.96

[40] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020. [Online]. Available: https://arxiv.org/abs/1802.03426

[41] ——, "How umap works," 2025, accessed: 2025-03-15. [Online]. Available: https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

[42] A. R. Wilcox, "Indices of qualitative variation," Tennessee, Technical Report, 1967, accessed: 2025-03-18. [Online]. Available: https://digital.library.unt.edu/ark:/67531/metadc864459/

[43] T. Bonald, N. de Lara, Q. Lutz, and B. Charpentier, "Scikit-network: Graph analysis in python," *Journal of Machine Learning Research*, vol. 21, no. 185, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-412.html

[44] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, p. 2837–2854, Dec. 2010.

[45] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," 2011. [Online]. Available: https://arxiv.org/abs/1109.2378

A Additional Figures

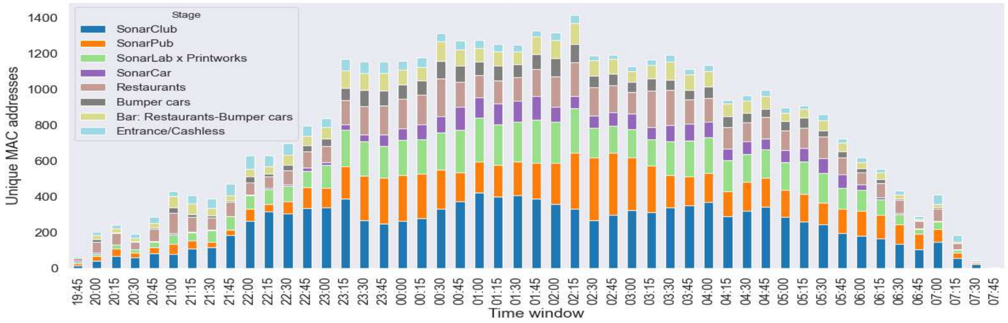


Figure A.1: Unique MAC addresses per 15-minute window for night 1.

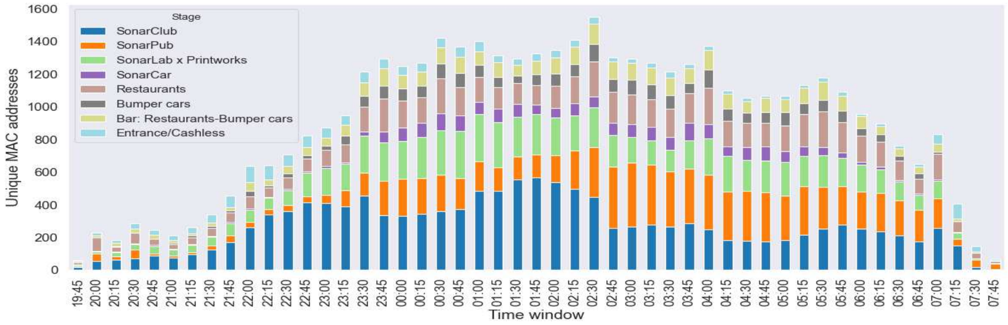


Figure A.2: Unique MAC addresses per 15-minute window for night 2.

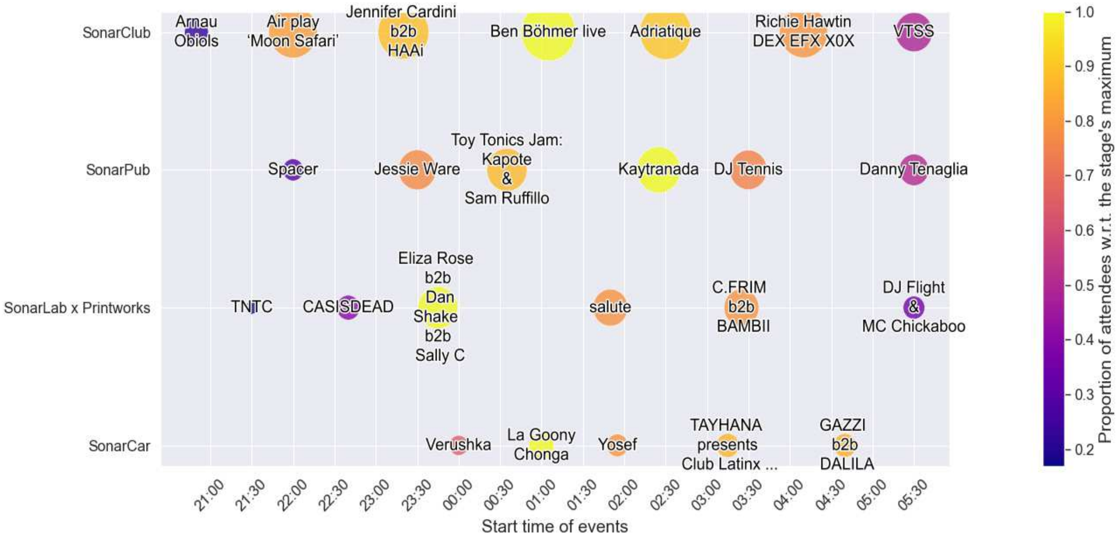


Figure A.3: Lineup and attendee count by event for night 1. The size of the bubbles represents the total number of unique MAC addresses detected at each event. The color indicates the proportion of devices at each event relative to the maximum for each stage on that day.

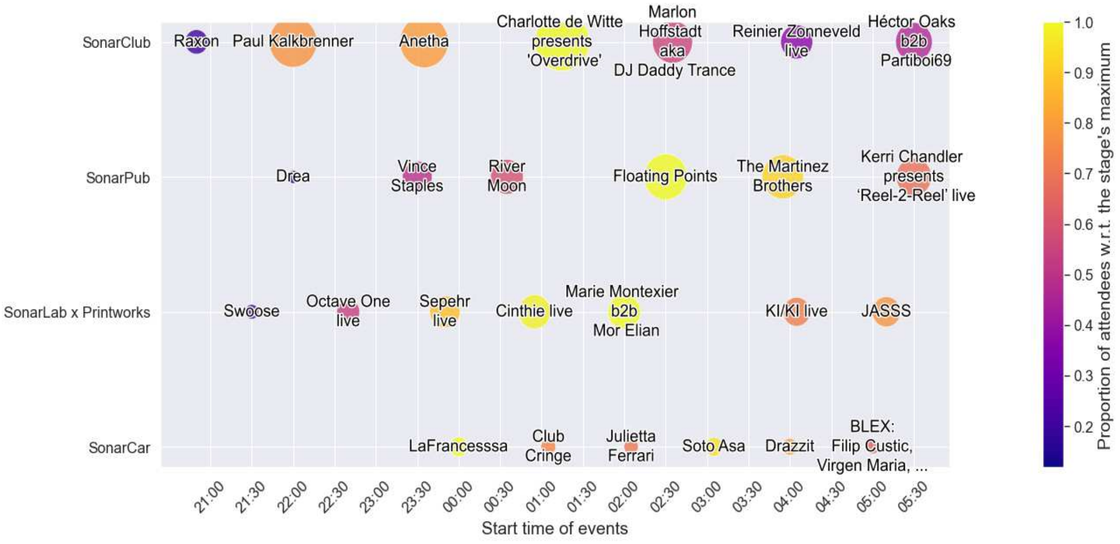


Figure A.4: Lineup and attendee count by event for night 2. The size of the bubbles represents the total number of unique MAC addresses detected at each event. The color indicates the proportion of devices at each event relative to the maximum for each stage on that day.

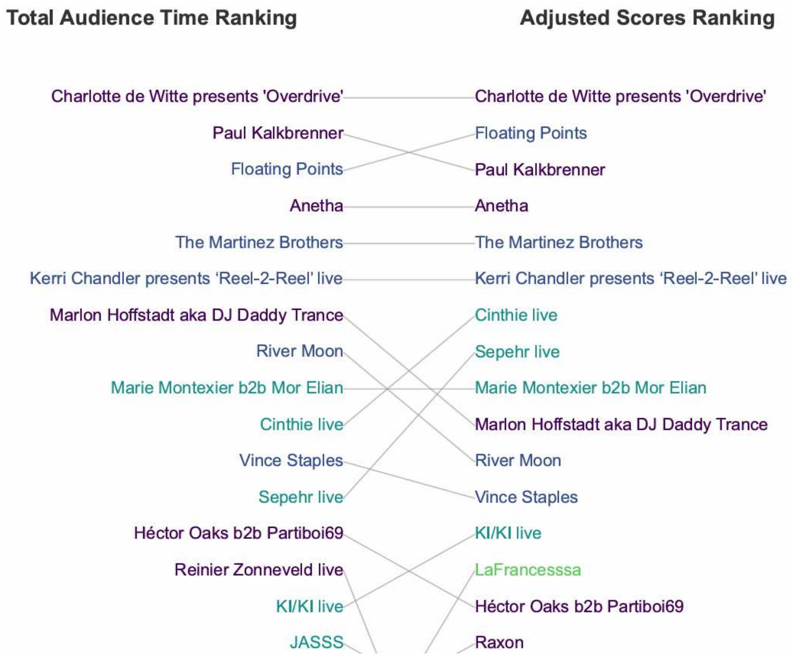


Figure A.5: Event rankings derived from implicit scores (cluster 1, night 2).

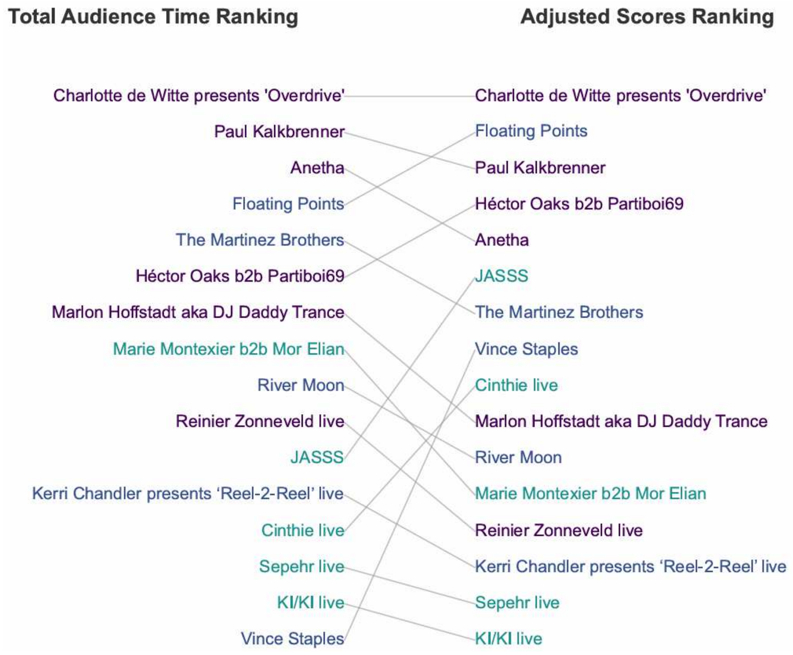


Figure A.6: Event rankings derived from implicit scores (cluster 2, night 2).

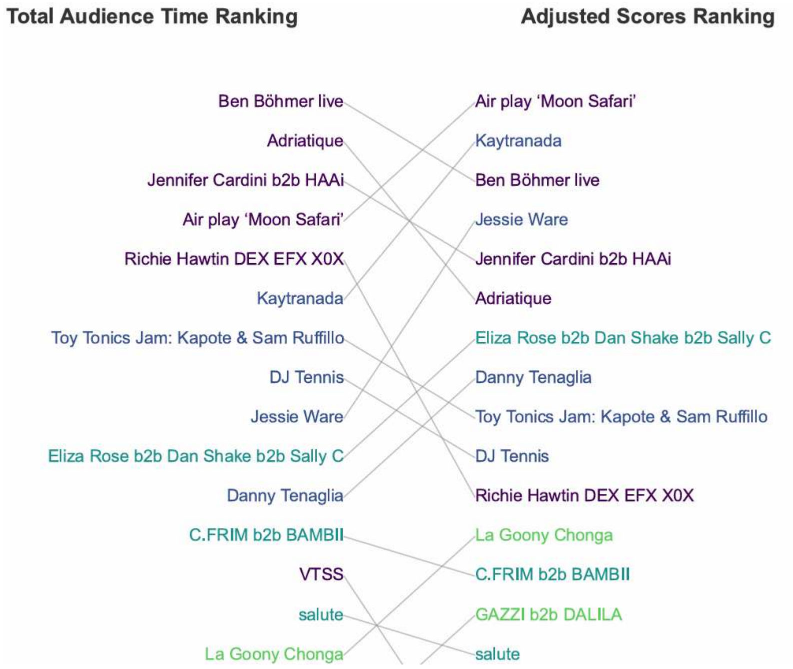


Figure A.7: Event rankings derived from implicit scores (cluster 1, night 1). Only the top-ranked events are displayed, with different colors indicating various stages.

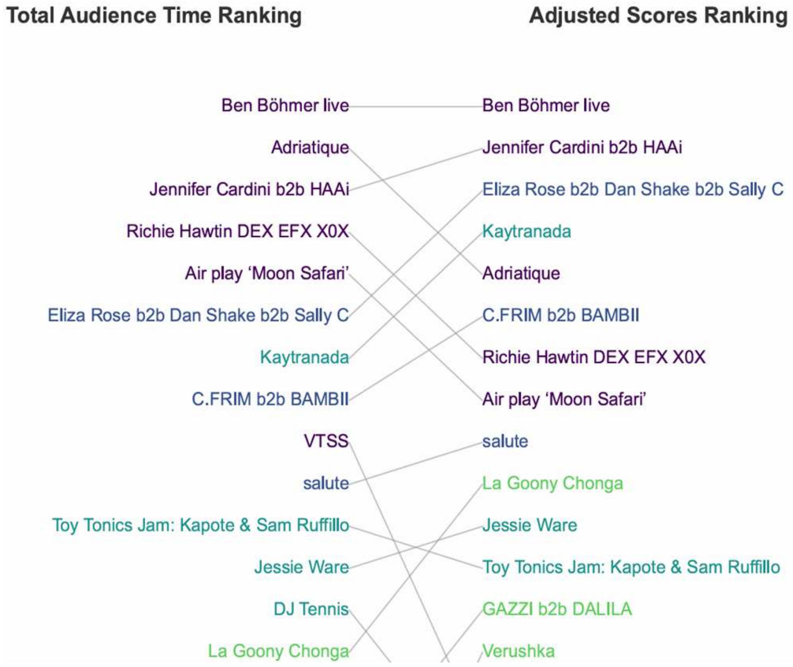


Figure A.8: Event rankings derived from implicit scores (cluster 3, night 1).

Acknowledgments

I would like to express my gratitude to the people who have supported me through this process.

Regarding the thesis itself, from Barcelona Supercomputing Center, I want to thank Patricio, for his kindness and knowledgeable guidance; Roger—who practically acted as a second co-advisor—for his constant support and advice; and Fernando, for his time and valuable insights. Also, I thank my supervisor, Professor Rossi, for his respectful character and timely assistance.

On a more intimate level, I want to thank my parents and my brother for having my back in every situation, always providing me with the love and support needed to achieve my goals. I am fortunate to never have to worry about this.

A special note of gratitude goes to Estefanía, who accompanied me through the ups and downs of this entire experience, helping me turn things around whenever the situation felt overwhelming.