Document downloaded from:

http://hdl.handle.net/10251/183670

This paper must be cited as:

Marchuet, D.; Palanca Cámara, J.; Botti, V. (2021). Wavelet Frequency Tensor applied to temporary environments in discrete spaces to obtain mobility patterns. Use case in the detection of routes in a territory. Engineering Applications of Artificial Intelligence. 106:1-16. https://doi.org/10.1016/j.engappai.2021.104507

The final publication is available at

https://doi.org/10.1016/j.engappai.2021.104507

Copyright Elsevier

Additional Information

Wavelet Frequency Tensor applied to temporary environments in discrete spaces to obtain mobility patterns. Use case in the detection of routes in a territory

Dani Marchuet, Javi Palanca, Vicent Botti

Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politècnica de València
Camí de Vera s/n, 46022, València, Spain.

Abstract

The use of frequency tensors for the representation of discrete time series information through wavelet transformations offers a methodology that allows the application of classification methods that result in the detection of mobility patterns in a geographical area. The use case focuses on a territory from which the geolocation of anonymised mobile device identifiers is extracted to aggregate its frequencies in order to ensure personal privacy. Furthermore, the kind of datasets we used in our study were treated with other methods to ensure that the identifiers for a same user change over time. Nowadays, this form of data treatment is implemented using a continuous process in smartcity platforms, first by anonymizing and second by changing the seed of the encryption method. Where the suggested technique is applied, a cluster analysis is performed, by which subsets or routes of the movements between the different points studied are obtained.

1. Introduction

Mobility patterns are extremely useful information because they allow information to be extracted regarding the movements of people in the studied geographical space. Data capture and analysis provide a better understanding of the movements and their possible associated impacts in order to achieve more sustainable mobility, better public transport plans, and improved efficiency of infrastructure investments. To do this, it is important to have data that allows for the analysis of movement. This study proposes a technique that can be extrapolated to any geographic area that contains frequency information associated with its points of interest. Taking as a basis the Wavelet transformation processes, a frequency tensor is generated between the different points of the

Email addresses: danimarchuet@gmail.com (Dani Marchuet), jpalanca@dsic.upv.es (Javi Palanca), vbotti@dsic.upv.es (Vicent Botti)

discrete spatial environment considered, and a transformation is proposed for its use in further analysis. Additionally, several methods are proposed that try to concentrate all of the information to generate extracts or feature variables of the tensor, for use in unsupervised learning methods to obtain the mobility patterns.

This technique can be generalized for its application in the extraction of mobility patterns in a cloud of elements belonging to any geographical area that has a geometric interpretation and whose relationships have associated time based information.

In the case study, we analyze the detection of mobile devices delimited in a time interval within a geographical area. These measurements are transformed into movements between pairs of places. The proposed method handles massive mobility data and obtains strong cluster structures. These cluster structures can be used to obtain mobility patterns between a set of points of interest.

The rest of the article is structured as follows: In Section 2, a review of previous works in the literature is presented. In Section 3, the proposed methodology for conducting the mobility analysis is laid out. In Section 4, we explain the proposed methodology for obtaining routes, and all of these steps are applied in a use case in section 5. Section 6 presents the experimentation and, Section 7 presents our conclusions.

2. State of the art

In order to have a vision of the current state of the art, regarding the framework of our research, we have taken into account two fundamental axis on which this type of study focuses: on the one hand how methodologies are used to extract characteristics from time series, and how mobility data is currently analyzed. Finally, we present the contribution of our study using the combination and integration of both axes.

2.1. Extraction of feature variables from time series

When we try to analyze mobility behaviors, movements can be viewed as a function that applies to the time series of detections between different locations. These time series are data that are collected over a period of time in which these observations occur, feature variables are then extracted from the time series for later use in classification processes, such as obtaining patterns or clusters of routes. These features extraction methods have been approached over time from different angles, including the use of Wavelet transformations of continuous and discrete signals [24]. These processes of extraction of variables based on transformations are usually represented by "kernels" in the application of convolutions. This process is usually seen in methods applied in the field of signal processing.[12]. And can be used to make predictions using unsupervised learning algorithms.

Analysis using wavelet transformations is a widely studied technique that is applied to the environment of sound filters in analog signals, image processing, and other image preview and compression methods. It also allows the framing of large time intervals into segments where greater precision for low frequencies is required, or into smaller segments where information with a high frequency is required, facilitating the subsequent analysis of the time series.

The pre-transformation signal can be viewed as a function whose resulting values are the frequency amplitudes for each moment in time.These filters can be iterated in multilevel filters and should be expanded based on signal we study. Multilevel decay is known as the Wavelet decay tree branched out according to the nature of the signal to be studied, in order to achieve the optimum decay [13]. This will obtain feature variables to be used in subsequent clustering processes [15].

2.2. Mobility analysis

Studies addressing mobility analysis and pattern extraction have the identification of frequent locations and the determination of stops and trips [7][2] as the common denominator. All of them start from minimum times and other thresholds that are previously established through the use of statistical indicators of specific cities and territories in order to obtain the movements. The methodology of this study greatly affects the involuntary discrimination of short stops and overlooks long distance trips (or routes) that have stops or intermediate passing points. By establishing parameters in these previous cited methods such as minimum travel times and other thresholds; this causes the set of obtained movement patterns to be very limited. The main hypothesis of all these studies is that individuals stop for a certain amount of time in a place to carry out activities such as visiting casual interesting points or waiting for transport links. Also, the trips are made between one activity and another, then the set of movement patterns obtained is very limited. However, this minimum time of activity must be established, and there is a disparity of criteria regarding the minimum time value that a stop should have [9, 5][8]. There are also approaches that rely on official reports on the time slots in which travel occurs [1] based primarily on probability functions.

Another line of study is based on measuring the entropy of individual trajectories, [23], breaking entropy down into three types: those that are random, linked to the individual's own habits, and the time entropy (this is depends on the frequency of visits and also on the order in which each point is visited and the time spent at each one).

Some authors use predictive methods (e.g. WhereNext [14]) which are based on anticipating the next location of a moving object from a preconceived outline of trajectories representing the habits and movements between places that are frequently visited during the trips. A decision tree called a T-pattern Tree is modelled by supervised learning methods. Most of the studies are applied to specific means of transport and take into user demographic characteristics as well as other zonal customs in the journeys [17].

Therefore with regard to the state of the art, we have a methodological background for the extraction of mobility information through the use of unsupervised learning algorithms and also the technique of time series analysis through wavelet transformations which is widely used. However these two fields of study have not been combined to obtain a methodology that alows feature variables to be extracted from a time series using the concept of what a wavelet transformation is. In this study, we propose an adaptation and combination of the two techniques for the purpose of extracting mobility patterns using unsupervised learning algorithms. This study addresses the adaptation and use of the two concepts to be applied to the extraction of mobility patterns in an innovative way.

3. Proposed methodology for mobility analysis

The technique we propose (Figure 1) includes an extraction of the feature variables of the time series of mobile device geolocations through the application of a wavelet transformation for subsequent treatment by clustering techniques. The purpose of the method is to obtain generalized patterns of movements. Unlike techniques cited by other authors, our method does not establish parameters that define the stops of individuals, nor does it take into account specific sources that characterize the custom and habits of the geographical area. It does not require the maximum time it takes to go from one point to another, and it does not rule out possible intermediate points. Our method allows independence of the possible meanings of the activities or visits, or the possible reasons inherent to the movements that ultimately translate into routes.

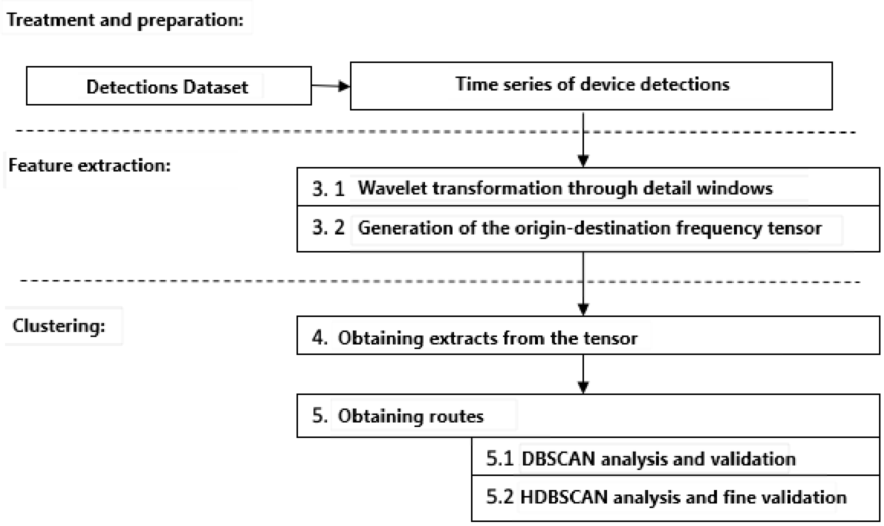

Treatment and preparation:
Detections Dataset
Time series of device detections
Feature extraction:
3. 1 Wavelet transformation through detail windows
3. 2 Generation of the origin-destination frequency tensor
Clustering:
4. Obtaining extracts from the tensor
5. Obtaining routes
5.1 DBSCAN analysis and validation
5.2 HDBSCAN analysis and fine validation

Figure 1: Outline of the methodology

Below we present each of the feature extraction and clustering methods in detail.

3.1. Wavelet transformation through detail windows

The continuous Wavelet transformation decomposes the signal into translated (time) and scaled versions of the original Wavelet that is used as the mother Wavelet to perform the transformation. There is a large group of families of commonly used Wavelet functions such as the Haar, Daubechies, Biortogonal, Coiflets, Symlets, Morlet, Mexican Hat, and Meyer, among others.

For each value of the S scale, the signal to be analyzed is multiplied by the selected mother Wavelet and integrated in the time space under study. The result of this integration is multiplied by the inverse of the square root of S to normalize the magnitudes. The result is the value of the Wavelet Transformation for that scale value. Conceptually, the wavelet transformation is the result of the product of the time series itself with the scaling and the transformation function $\omega(x)$ that has a given kernel or base function as its origin. As can be seen in the Equation 1, the scaling and the transformation actions are defined by the scaling variables "s" and the parameter "b". The variable "s" adapts the level of granularity of the signal (which can ever reach the maximum level of detail determined by the frequency data); the parameter "b" determines the location of the section of the function [26].

$$Wf(s,b) =< f,\omega > (s,b) = 1/s \int \Omega(x-b)/s) \tag{1}$$

Logically the result will depend on the selection of the mother Wavelet ($\omega(x)$) and the choice of the orthogonal base. This will be essential in the process of determining the best decomposition. This process is based on how the transformed coefficients will provide more information about the original time series.

If we want to apply the Wavelet transformation to a numerical data series, a discrete transformation is implemented [22]. Our study focuses on establishing analogies with the discrete transformation, which is usually performed by applying level filters.

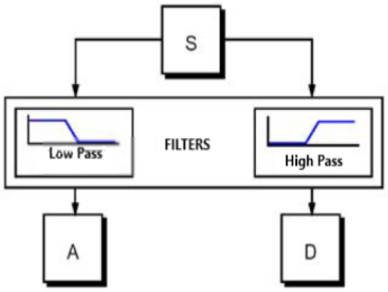


Figure 2: Wavelet Discrete Transformation Filters

According to these studies. Most signals have a low-frequency component that characterizes the signal that breaks down using the "Low Pass" filter, while the high-frequency components incorporate more specific characteristics ("High Pass" filter). Schematically, Figure 2 has as input S (the signal to be analyzed) as input, A is the "low pass" filter output, and D is the "high pass" filter output.

3.2. Generation of the frequency tensor

The filters that we apply in this work are based on time windows that accumulate the frequencies of the geolocated entries, which are produced by the movements during that interval.

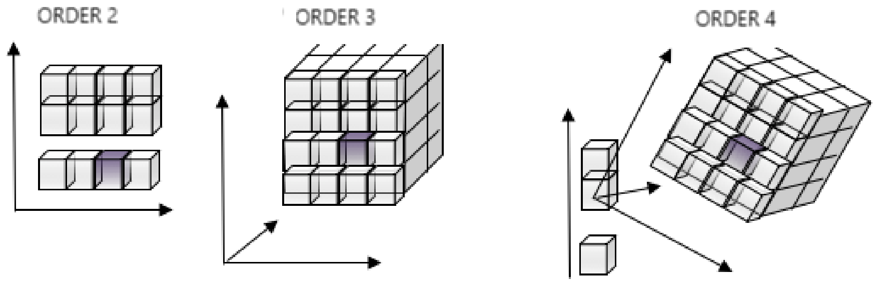


Figure 3: Tensors of order 2, 3 and 4

We start from the device geolocations, and after elaborating the datasets containing the information on the movements, we propose a methodology to extract their frequencies following the analogy of the temporal analysis of discrete signals by means of wavelets. This methodology results in an origin-destination frequency tensor whose components make up a multi-dimensional matrix. In order to identify a frequency value within this multi-dimensional matrix a tensor of order 4 will be formed. This tensor is composed of the following indices: the source identifier, the target identifier, the frequency window and the output moment (Figure 3).

3.3. Generation of tensor extracts

As in the case of convolutional neuronal networks, we propose techniques to extract the information contained in the proposed tensor, to do this we use a series of techniques that allow the extraction of feature variables from the tensor.These feature variables are inputs in the classification process we used later.

Figure 4 shows an example of extracting values from an order-4 tensor in a quadratic structure. There we apply two functions to the tensor in order to obtain a quadratic structure. At the end of this process, the different proposed methods obtain a quadratic structure, composed of frequency-based feature variables that are extracted for each methodology in order to obtain these accumulated values.

The different versions of the obtained dataframes are used as input for the proposed classification algorithms. After evaluating the metrics of the structures obtained, we select the feature extraction method that generates the highest

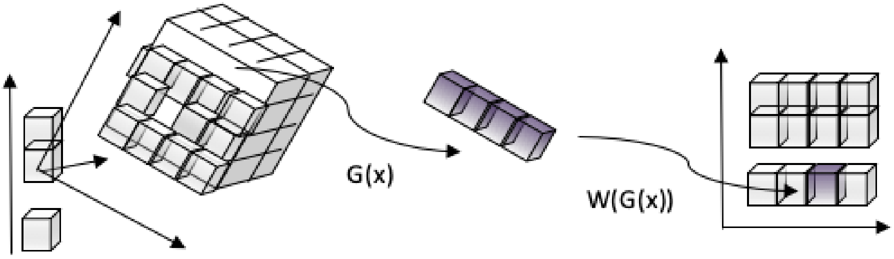


Figure 4: Tensors of order 2, 3 and 4

performance in the clustering process. Then we apply it to each obtained dataframe to evaluate and compare the fitting metrics of the cluster structure obtained in this process.

4. Obtaining routes

Obtaining routes from the point of view of classification has been proposed by different authors using clustering algorithms [25][10]. In this work mobility patterns are obtained two phases: in the first phase we use the DBSCAN algorithm, in order to select the best scalar vector. In the second phase, we apply HDBSCAN to obtain the final routes.

The DBSCAN algorithm is based on densities [6] and locates clusters by distinguishing betwen high density and low density areas. Unlike other methods, this algorithm can detect non-convex clusters, and it is not necessary the number of clusters to be obtained as a parameter. Nor is it necessary to know previously (or to intuit) the number of reference clusters. The concepts on which this algorithm is based indicate the number of occurrences for an element to be part of a density kernel. This is established in the "Samples" and "epsilon" ("eps) variables. This variables specify the minimum density obtained that each cluster.

In this work, we evaluate the different proposed alternatives according to the methods considered for obtaining the scalar vectors or pooling methods. We also carry out a comparative study of the resulting structures. Finally, we calculate the silhouette coefficients [18] for each structure in order to select the one that provides the greatest consistency and thus validates the results obtained.

The evaluation of the coefficients was carried out following the interpretation of the silhouette indexes based on to the intervals shown in Table 1.

A classification process is launched for each of the data structures of the origin-destination tensor extracts obtained in windowed wavelet processing.

The results or groupings obtained by the DBSCAN algorithm are based on point density estimations around the point established by the *Epsilon* (ϵ) parameter, which defines the grouping radius with respect to the density core point and the number of neighboring points for an element to be considered a density core. The *Samples* parameter determines this density, which is the

Silhouette Index Range	**Interpretation** [11]
0.71-1.0	Strong structure
0.51-0.70	Reasonable structure
0.26-0.50	The structure is weak and could be artificial
< 0.25	No structure

Table 1: Interpretation of silhouette indices according to intervals

number of neighbouring points to differentiate these areas from others of lower density. All of the points belonging to this area are considered density cores, on which this rule is applied several times, so the method iterates and agglutinates the points in clusters. The areas grown as reachable points are incorporated.

In the second phase of clustering, once we have detected the best alternative frequency tensor extract using DBSCAN, we apply the method *HDBSCAN* [3] to refine and provide the final results with the best precision. HDBSCAN classification technique is based on the DBSCAN algorithm, wich varies the values of $Epsilon(\epsilon)$ and integrates the results to find a classification that provides the greatest stability over *Epsilon*. This allows having dense clusters with different densities, unlike DBSCAN, which discards the areas of points with low densities. By using branching and pruning methods, this algorithm reduces the computational complexity and also has the advantage that only the minimum cluster size is required as an input parameter. This way, the range of possible variations in the two initial parameters of the DBSCAN is reduced and the computational complexity for executing each one of the tests is also reduced. This feature allows programming a range of variation using only one parameter to run the executions, and it also achieves better results for frequency distributions in discrete spaces.

5. Use case

This section proposes a use case where the proposed methodology is applied to the geolocations of mobile devices in a specific geographical area. We explain in detail the treatment of the frequencies obtained in order to generate a set of origin-destination matrices that are capable of concentrating all of the information without having to know in advance or parameterize what a stop or activity is, or take into account specific characteristics of the area of study in order to identify trips (associated with routes) and stops (depending on activities).

Explaining the proposed methodology applied to the use case:

The method proposes to make the obtaining of the movement patterns more flexible as alternative for obtaining and transforming the frequencies of the movements. This is generalizable to other possible cases of frecuency analysis obtained from time series.

In our use case, we tackle the decomposition of the frequencies, given by the instants of time on which our analysis is focused. We can extract the corresponding amplitudes from the dataset containing the movements for each frequency considered as a multilevel filter. The objective of the Wavelet transformation that we propose is to prepare data that will facilitate a later classification analysis of the frequencies observed among the discrete space of points that are studied. Our case contains frequencies of the number of first-degree movements, between fixed origins and destinations, inside the detection points (or antennas) deployed in the geographical area included in our dataset (Dakar territory). However, other datasets with user detections associated with points of interest can be used. The user id must be anonymized and changed periodically during the scope of the dataset [16]. Thus, an id references the same user only during the same period.

The information on the movements between places or points of interest in a territory is gathered in mobility matrices, whose content is the accumulated frequencies for a given moment of time between all possible origins and destinations that are associated to the use case geographical area.

We apply the technique explained about the concept of Wavelet transformation. To do so, we establish a parallelism with the analysis of analogical signals in frequencies and amplitude, with the aim of being able to have in a single object all the information about the behaviour of a discrete function, which in our case will be the number of movements that occur between two points of interest in a given time interval by unique users.

We apply 20 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, and 12 hours as spectrum "views" that vary the level of detail, from those with a basic level to those with a precision established by the accuracy of the original dataset. The use case dataset is in a discrete environment. We select views of this kind according to the frequency levels of the original signal (Figure 5).

In the case of the Wavelet transformation we use, each base function is materialized in a layer that represents the value of the expansion coefficient (as the accumulated number of movements calculated). This is similar to the low-pass and high-pass filters, but with multiple filters as a function, which allow views of the data with different levels of detail. In this way, a series of layers is generated by levels of detail that are defined when the transformation function is applied.

5.1. Discreet wavelet transformation application

5.1.1. Extraction function for movements between two points

In order to extract the movements between i and j in a specific time interval (Figure 6), we define the range of movements between i and j in the v time interval as the number of unique users who went from i at the beginning of the interval to j at the end of the $(t + v)$ interval. This reasoning is made on the first-degree movement datasets $M(i, j, t)$. This include any type of movement between two points of interest(including movements that pass through intermediate points), since we only consider the origins and destinations made by unique users.

In order to use the d method, it is necessary to perfectly know the feature variables selected and the distribution of their points in space in terms of the distances between them.

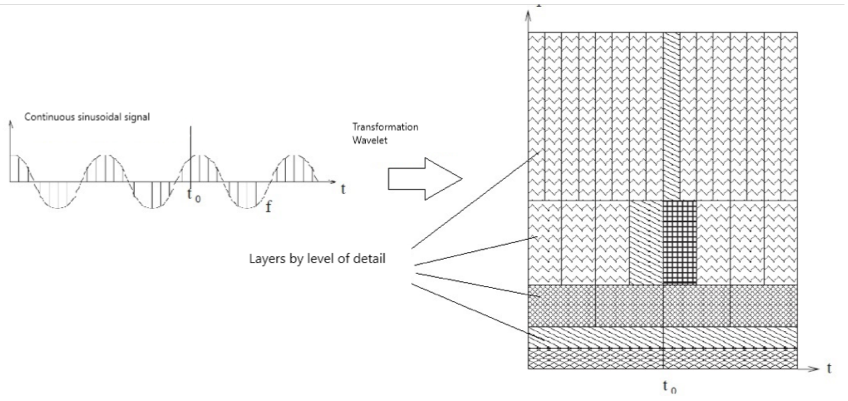


Figure 5: Schematic view of a wavelet transformation

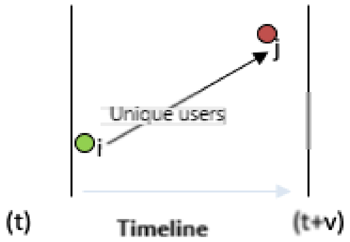


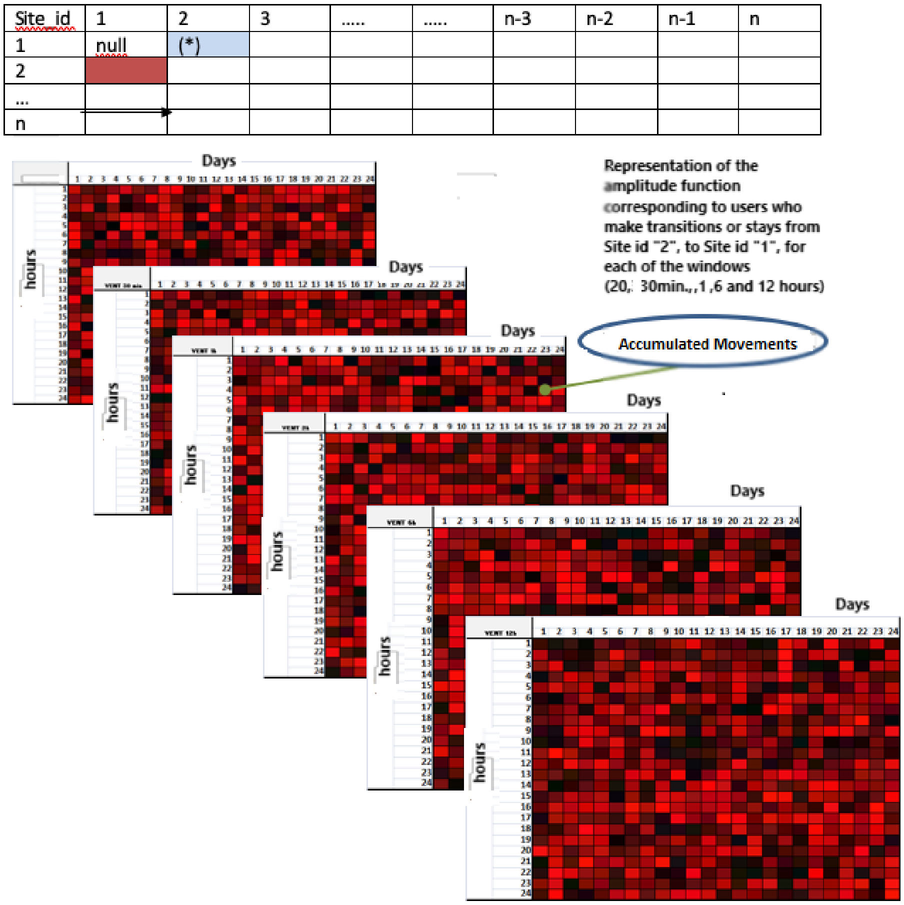
Figure 6: Representation of the range of movement in $t + v$ between the points i and j

5.2. Generation of the main origin-destination tensor $(DT(i, j, t, v, f))$

Obtaining the number of movements detected is based on a process of counting movements. To do this, we iterate the first-degree movement dataset, extracting all of the movements that have been accumulated in each of the cells between each pair of points of interest (antennas) i, j,and for each "wavelet zone". This is generated by iterating the indices, i, j, for each of the considered windows of 20 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 12 hours, with an hourly granularity in ranges of 24 days. We obtain the values by applying the tensor extracts generation process (defined in Section 3.2), with the aim of composing each cell of the source-destination tensor, which makes up the proposed wavelet structure, we obtain a single tensor $(DT(i, j, t, v, f))$ of origin-destination for a temporary stretch "t" of 24 hours for 24 days (Figure 7).

Each cell of the origin-destination matrix (Figure 7) is composed of six 24-hour matrices for 24 days. Forming orthogonal matrices facilitates the transformations that we propose and that serve as input for the subsequent classification analysis.

The period sections or windows have a variable scale granularity (20 min, 30 min, 1 hour, 2 hours, 6 hours, 12 hours). This granularity is expressed in the hourly intervals within each day that contain the number of exits that occur from



Site_id	1	2	3	.....	.....	n-3	n-2	n-1	n
1	null	(*)							
2	🟥								
...									
n									

Representation of the amplitude function corresponding to users who make transitions or stays from Site id "2", to Site id "1", for each of the windows (20,: 30min.,,1,6 and 12 hours)

Figure 7: Graphic representation of the proposed Wavelet Transformation in the t interval (24 hours x 24 days).Every cell of origin-destination matrix is decomposed in six frequency matrices

the initial node to the end node, whose movement duration is less than or equal to the length of the window represented. The study was carried out by adding two of these 14 possible groups, thus covering a time span of 48 consecutive days (i.e. adding the frequencies of two 24-day periods).

The scope applied in our use case is a zone that contains a total of 492 nodes. Therefore, we will have a tensor of 492 rows x 492 columns x 6 windows x 24h x 24 days. It results in a dataset whose upper bound is 242,064 three-dimensional elements. In this way, the computational needs of subsequent treatments are reduced and limited. Furthermore, it must be taken into account that when going through the tensor, in subsequent processes of generating new extract type data structures, we eliminate those that do not have any movement between the corresponding origins and destinations.

Both the sections or windows and definition of the temporal scope to carry out the aggregations should be adapted to the accuracy of data timeline and the time required to cover the maximum distance between the two furthest points of interest of each study. These parameters will be affected by the frequency and precision of the samples. In our case, the detections are recorded every 10 minutes, which limits the minimum interval of the defined section or window and must be equal to or greater than the precision of the records contained in the dataset under analysis.

The territorial scope of application where the study is carried out defines the maximum time window in which our model analyzes the movements. Time windows that exceed the time necessary to travel the maximum distances between the most distant points of interest will not be considered, since we know in advance that their frequencies will be zero. For example, in a small territorial area, it may not make sense to consider trips of more than one day, if for example, longer trips can be covered on foot. To establish this, we take into account the means of transport to be included and the maximum distance defined between the furthest points in the geographical scope of the analysis. Depending on the type of means of transport to be analyzed in the study, the users will have a maximum time to cover the distance between the furthest points of the territory. This will implicitly define the magnitude of this window. It should be taken into account that this window limits the duration of the maximum route to be considered in the analysis. In this case, routes longer than 12 hours are not considered. Within this time, possible stopovers or intermediate visits to be made on the routes between two points of interest are included.

The matrices generated can also include the classifications of the days of the week, or they can be split into sections (holidays or working days). Even though this would increase the dimensionality of the matrices and thus the computational complexity of the subsequent treatments. It will most likely enrich the results of the classification stage.

5.3. Description of the stages

This method has the following 6 stages:

1. Analysis of the dataset to be used in the study. Its characteristics, precision, and granularity will limit the selection of parameters to be used in the transformation.
2. Definition of the time frame and resolution space V_0. We establish the interval of the study. For the execution of the model of our use case we selected 48 consecutive days from the detection data sets, in which we add the frequencies and perform the transformations with a granularity of 24 hours.
3. Loading of the datasets containing the geolocation of the points of interest and the union of these datasets for the selected interval. This is called the detection vector ($D(i)$).
4. Selection of the ranges of the windows "v". To define them, the minimum and maximum distances, route times, and means of transport between the

different locations of the points of interest to be considered in the study must be taken into account. In our case, the trips take place within a single province, which is why we do not consider windows greater than 12 hours. This scale will have to be adapted to each use case and will also depend on the temporal granularity of the obtained dataset. When this type of filter is applied to our set of data on movements, the first level of the filter would select intervals of 20 minutes, the second level would select 30 minutes, the third level would select 1 hour and so on, until we reach the 12-hour filter, which is the one that groups movements with the lowest level of detail.

5. Application of the function to obtain first-degree movements ($E(x)$). After its application on the detection vector ($D(i)$), we obtain the first-degree movement vector $M(i,j)$, which contains the movements between all of the origins (i) and destinations (j) contained in the detection dataset.
6. Application of the wavelet transformation function ($Fw(M,i,j,t,v)$) on the first-degree movements ($M(i,j)$): the wavelet function provides the accumulated value of the number of unique users that have moved from point i to point j for each of the windows or levels of detail. Its parameters or input variables are: $txnd =$ timeline (24 hours x 24 days) and $V=$time windows (every 20 minutes, 30 minutes, 1 hour, 2 hours, 6 hours, 12 hours). Our transformation function on the first-degree movement vector you get $a=$amplitude (number of unique users), which tells us the number or accumulation of unique users detected in the first-degree movements dataset $M(i,j)$ within the treated time interval. In our case this is the number of movements corresponding to the window considered for each scale. This is how the filters of the proposed discrete transformation are constituted. This function calculates the accumulation of the movements of each unique user who has first-degree movements between the origins and destinations of the dataset of first-degree movements within each of the windows in a grid of 24 hours for 24 days. It is not necessary to standardize the values obtained since all of the variables used in the study are frequency ranges expressed by integer values that indicate only the number of detections and accumulated displacements in a time interval, so their magnitudes are homogeneous and comparable.

5.4. Synthetic expression of the process

We define the first-degree shift extraction function as $E(D(i)) \equiv M(i,j)$, where $D(i) \equiv$ the vector of user geolocations at point i, which contains the timestamp of the detection and the unique identifier of the user.

For each detection of the detection vector, we take as origin and extract the movements for each possible destination for each u, or single user. $M(i,j,t)$ is the vector of first degree movements made by users between the points of interest i and j at the instant of time t, who leave i at the moment t and are detected to reach j. It only includes direct trips between two points of interest. Thus, we define the wavelet function (Fw) as Equation 2:

$$Fw(M, i, j, t, v) = DT(i, j, t, v, f) \tag{2}$$

$$f = \sum^{u=1..n} (M(, i, j, t + v)) \tag{3}$$

where $i \equiv$ point of interest origin, $j \equiv$ point of interest destination, $h \equiv$ hours (from 0 to 23 hours), $nd \equiv$ number of days considered (day 1...24 days), $t \equiv$ moment of time $(h+nd)$, $v \equiv$ window or time interval (of amplitude $\equiv$15 minutes, 30 minutes, 1 hour, 2 hours, 6 h., 12 h.), $u \equiv$ unique user of the dataset, $n \equiv$ total detections. Being $DT(i, j, t, v, f) \equiv$ movement tensor. Each f element of the tensor is the sum of the movements of single users who have left i at the moment t and have reached j during $t + v$ (in the time interval defined by the v window), expressed at Equation 3. This includes the accumulation of possible movements passing through other points of interest.

6. Materials and methods

In this work we have used the "D4D-Senegal: The Second Mobile Phone Data for Development Challenge" dataset. We had access only to this dataset but another dataset with similar structures could have been used, because the methodology and the model used is only based on the accumulated frequencies of mobility and the detections. We don't use another kind of variables associated with the territory, such as transport infrastructure variables or other geographical or demographic data. This is a type-2 dataset ("Fine-grained mobility") consisting of 25 datasets provided by Orange Senegal in csv file format. The dataset is composed of a user identifier, a timestamp, and a node identifier. Each of the working datasets corresponds to a period of two weeks. This study window accumulates approximately 44,400,000 records. The user identifiers of two datasets of different periods do not match. They contain location information that is associated with 1666 antennas from 320,000 random users. In Table 2 shows an example of the dataset along with the structure of each of the original files.

The methodology based solely on the frequencies of detections and movements allows the results to be independent of other variables intrinsically associated with a geographic area, which makes it easier to extrapolate to any other territory.

Figure 8 shows the granularity of locations guided by the distribution of antennas throughout the geographical area. We see that there are areas with a higher level of coverage and with a higher granularity, which in the case of Senegal are concentrated around Dakar.

6.1. Methods used on the detection dataset

We perform a process that includes all the steps until the frequency tensor is constituted. All of the mechanisms that have been developed to process the data and apply the proposal of this work are shown below.

user$_{id}$	timestamp	site$_{id}$
1	2013-03-18 21:30:00	716
1	2013-03-18 21:40:00	718
1	2013-03-19 20:40:00	716
1	2013-03-19 20:40:00	716
1	2013-03-19 20:40:00	716
1	2013-03-19 20:40:00	716
1	2013-03-19 21:00:00	716
1	2013-03-19 21:30:00	718
1	2013-03-20 09:10:00	705
1	2013-03-21 13:00:00	705

Table 2: Sample of the input data

6.1.1. Method used to obtain first-degree movements

The objective of the first method carried out on the dataset is to extract movements between the different points of interest. We call these movements first-degree movements since they do not include possible intermediate points and could be seen as atomic movements made by single users. Algorithm 1 shows the extraction of all of the movements for each dataset, according to the process of obtaining first-degree movements.

After execution, we obtain the first-degree movements. The input datasets

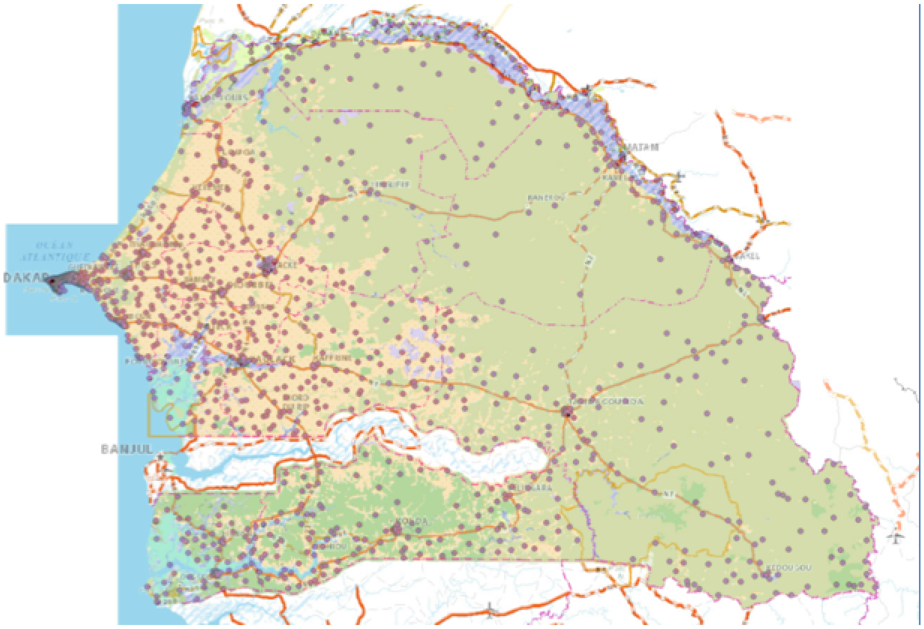


Figure 8: Distribution of antennas over the territory of Senegal

Algorithm 1: Extraction algorithm for all trips for each data set

```
Input: detection datasets
Result: first order movements
foreach dataset do
    dataset = sort(dataset, by=[user_id, timestamp]);
    last_user_id set to None;
    last_place_id set to None;
    foreach user_id, place_id, movement_data $\in$ dataset do
        if user_id is last_user_id then
            if place_id is not last_place_id then
                store_movement(movement_data);
            end
        end
        last_user_id set to user_id;
        last_place_id set to place_id;
    end
end
```

have been previously filtered, limiting them to the geolocations of the Dakar province instead of using the whole geographical area of Senegal. This way, we reduce the computational complexity and adapt the data load to the resources that are available for the present study.

6.1.2. Method used to obtain the frequency tensor

To form the frequency tensor vector, we carry out a series of four functions on the first-degree movements described below.

- Time window scrolling function:

Starting from the size of the windows to be processed (20min., 30min. ,1hour, 2 hours, 6 hours, 12 hours), we carry out an iterative process for each one of them in order to conform the spectrum of accumulated frequencies for each interval. To do this, we obtain the subset of movements that are within the selected window, taking the base timestamp specified by the day, hour, and minutes as an input parameters.

- Candidate acquisition function:

The purpose of this function is to extract the set of candidate movements for a specific time window (Algorithm 2). This function is broken down into the following 4 steps:

1. We calculate the final value that will limit the query of movements and will add the size of the window (num) to the initial moment, with $num \equiv (num\_day, num\_hour, num\_minutes)$:

$$tendday, tendhour, tendminutes \equiv (day, hour, minutes) + num$$

2. We select all of the users that have gone out on the day, hour, and minutes selected, and we execute the query on the movement dataset to obtain the outputs:

$$origins \equiv movements.query(day, hour, minutes)$$

3. We select all of the users that have arrived on the day, hour, and minute of the window's end time, and we run the query on the movement dataset to obtain the arrivals:

   crit_movs_destination $\equiv (day\_arrival \equiv tendday, hour\_arrival \equiv tendhour, min\_arrival \leq tendminutes)$

$$destinations \equiv movements.query(crit\_movs\_destination)$$

4. We obtain all of the movements in that time window:

$$movements \equiv innerjoin(origins, destinations, on \equiv user)$$

```
Algorithm 2: Candidate acquisition function. "getCandidates"
  Input: first order movements dataset, day, hour, min, num
  Result: movements_subdataset
  $tendday \equiv day + num.days$ ;
  $tendhour \equiv hour + num.hours$ ;
  $tendminutes \equiv minutes + num.minutes$ ;
  $origins \equiv movements.query(day, hour, minutes)$;
  $crit\_movs\_destination \equiv (day\_arrival \equiv tendday, hour\_arrival \equiv$
    $tendhour, min\_arrival \leq tendminutes)$;
  $destinations \equiv movements.query(crit\_movs\_destination)$;
  $movements\_subdataset \equiv innerjoin(origins, destinations, on \equiv user)$;
```

- Function to generate the frequency dataset:

This process (Algorithm 3) aims to synthetize the dataset containing the frequencies associated with each window for each moment of time. It achieves this by using the candidate procurement function (Algorithm 2) and then accumulating the frequencies.

- Frequency tensor generation function

We process the frequency dataset obtained in the previous section, we process it to generate the proposed frequency tensor using the following 3 steps:

1. We initiate the tensor that will accumulate the frequencies to zeros

   TensorFrecuencies=full((493,493,6,24,24),0)

2. We calculate the sums of the accumulated frequencies of the frequency dataset ("FrequencyDataset"), grouping by start, destination, num, departure day, departure time:

   Acumulated $\equiv$ FrequencyDataset.groupby

   (['origin','destination','num','dayDeparture','hourDeparture'])[frequencies'].sum

Algorithm 3: Obtaining the frequency dataset

```
Input: first order movements dataset
Result: frequency_dataset
foreach $day, hour, minute, num \in dataset$ do
    $AuxMatrix \equiv 0$;
    $candidates \equiv getCandidates(dataset, day, hour, min, num)$;
    foreach $candidate \in candidates$ do
        $origin \equiv candidate[origin]$ ;
        $destination \equiv candidate[destination]$;
        if $AuxMatrix(origin,destination) \equiv 0$ then
            $frequencies \equiv Count(candidates(origin, destination))$
             $AuxMatrix[origin, destination] \equiv 1$;
            // store in frequency wavelet dataset
            $frequency\_dataset.store(origin, destination, day,$
             $hour, minutes, num, frequencies)$;
        end
    end
    end
```

3. We go through the new "Accumulated" dataset and assign the accumulated values in the tensor.

As a result, we obtain a tensor that contains the accumulated frequencies for each origin, destination, window, and each instant of time considered.

The pseudocode is shown in Algorithm 4.

Algorithm 4: Generation of the frequency tensor

```
Input: FrecuenciesDataset, origin, destination, num, day, hour, minutes,
       frequencies
Result: Wavelet TensorFrequencies
$TensorFrequencies \equiv initializeTensor((492, 492, 6, 24, 24), 1)$;
$wavelet\_subset \equiv FrecuenciesDataset.groupby([origin, destination,$
 $num, day, hour, minutes])$;
$Acumulated \equiv wavelet\_subset[frequencies].sum$;
foreach $tuple \in Acumulated$ do
    $TensorFrequencies[origin, destination, num, day, hour] \equiv$
     $tuple[frequencies]$;
end
```

6.2. Techniques for reducing the size or generating extracts from the tensor

In this section we propose alternatives for the transformation of the frequency tensor to obtain a two-dimensional vector representative of it and that is more appropriate to be used in the classification process, since it is necessary to transform the tensor into a data structure that can be used as input by the classification algorithms.

The formation of the frequency matrix defined in previous method (at the Section 6.1.2) means that our information is stored in a tensor with the following characteristics:

$$origin\_destination\_Tensor \equiv (492, 492, 6, 24, 24) \equiv (i, j, window, hours, days)$$

In order to perform a clustering analysis of the tensor and obtain the patterns based on their frequencies in a feasible way, different transformations of vectors of numerical values must be used to express the content of the origin-destination matrix grids in a new data structure with a quadratic shape. This structure will be used in the proposed analysis phase.

The goal of obtaining a new dataset containing the tensor information in a new data structure made up of tuples is to adapt the parameter input and prepare the tuples to be able to execute the tensor load in the classification algorithms. These algorithms require datasets that are made up of rows and columns, where the rows are records and the columns are the feature variables with which they work to achieve the classifications. Data structures of this type are the ones that support most of the classification algorithms while reducing their computational complexity and allowing their widespread treatment in any type of process.

The objective of this transformation is to obtain input variables (or candidates) that are capable of being introduced in a classification algorithm, reducing the computational complexity in order to choose the best option from the proposed transformations. Below, we show the transformations that have been designed for this work, highlighting their advantages and operation. This method which is very similar to the "*pooling*" layers that are used in convolutional neural network models to obtain the representative characteristics from the convolutions of the previous layer, which are widely used in the analysis and prediction of signals or images.

To do this, we combine two ways of obtaining square submatrices of the tensor: one by means of a path of the tensor in which matrices of variable size are generated (Figure 9), and another in which the tensor is cut into fixed-size square matrices (Figure 10). In addition, we present three more functions to obtain scalars of the submatrices: one that composes a vector with the values of the means and the determinants obtained from the submatrices, another that compiles their maximum frequencies, and one that performs the summation of all of the elements of the submatrix. In the following sections we describe six alternatives to generate and extract a set of features starting from the frequency tensor obtained.

*6.2.1. Obtaining views of the tensor by means of determinants and averages of variable size submatrices (*Pooling*)*

The goal of this procedure is to obtain views with compressed information of the tensor to obtain representative values of two-dimensional zones or matrices. These zones are formed in square matrices in order to be able to carry out the calculation of determinants. The absolute value of a determinant can be seen as the measure or multiplication factor with which the matrix expands or contracts space. If the determinant is zero, then the space contracts completely in at least one dimension, losing all its volume. If the determinant is 1 then the transformation preserves the volume, which is the reason for initializing a value that expresses the magnitude of the matrix at the beginning. The calculation of the geometric average of the elements of the matrix expresses a weighting of all of the elements that make it up. The proposal aims to generate a vector that contains a set of scalar variables that made up of means and determinants from different areas of the tensor that are defined by the submatrices and the path proposed in Figure 9.

The methodology for obtaining these zones or submatrices is defined by the path and the direction of the arrows in Figure 9.The matrices obtained have to be squared in order to calculate their determinants. The determinants and averages are calculated according to the order of the arrows in the diagram.

The goal of the transformation proposal is to generate square matrices of different sizes on which to make calculations and thus extract the characteristics of determinants and averages of each sub-area or square matrix. We also present a method for choosing the area of these submatrices that vary in size. The choice of the zones is made by means of a route, which takes into account the hourly dimensions of the matrix the aim of which is to vary the selection of the zones according to the window in which we are making the calculations. Thus, in the 20-minute window, we begin to grow the dimension of the submatrices formed from a 2x2 dimension (avoiding the 1x1 matrix), calculating the determinant and the average of the matrix formed by the frequencies of the first two hours of the day of the first two days. Then we calculate the 3x3, 4x4, 5x5, 6x6, and soon until we reach the 24x24 matrix, which will include the 24 hours of the day of the 24 days of the matrix. By defining this path, which is the one shown by the arrow in Figure 9 for the values of the frequency matrix of the 20-minute window, we understand that the magnitudes of the calculations of the determinants and the averages of the last few hours will be much larger than the one calculated for the first ones because the square matrices have a larger dimension. That is why, for the matrix composed of the 30-minutes window, we reverse path. We start to generate submatrices for the last hours of the last day of the dataset, going towards the first hours of the first day.

In summary, the approach proposed in Figure 9 aims to group areas of different sizes and in a variable and incremental way. The determinants are grouped and calculated according to the arrow in the figure, which, varies its direction depending on the areas. In the first zone defined by the matrix of frequencies of 24 hours x 24 days, the first submatrix that is formed is that of 2

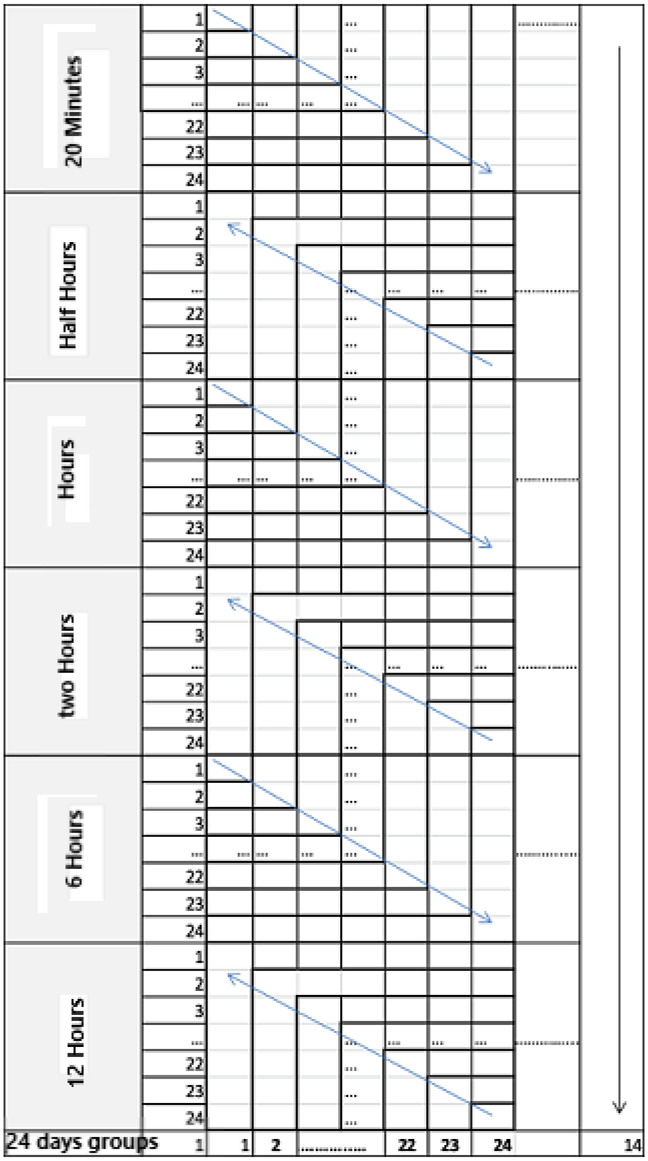


Figure 9: Variable window pooling method with path inversion. The arrows indicate the direction in which the sub-matrix calculations are performed. We apply the pooling method proposed by leading this direction, and extracting features by accumulating frequencies to the square marked in the route

hours x 2 days, then that of 3 hours x 3 days, and so on until (nxn), which would be that of 24 hours x 24 days. During the first grouping, we start with the first window of the tensor (20 minutes). The variables that are accumulated bring together the frequencies of the movements of a square matrix of 2h x 2 days. We calculate the two variables for each generated submatrix: the determinant and the average of that grid. The next pair of variables is formed by the accumulation of movements during the first three hours and the first three days, and so on until we reach the determinant that binds all of the hours of every day (24x24).

In Figure 9 the selection grouping method to extract the frequency characteristics of the tensor is shown. To do this we flatten the tensor into six matrices expressing the maximum duration of the displacements per hour for 24 days. This is done to propose a procedure based on the grouping of orthogonal sub-matrices to be able to perform calculations on them. In this case we propose a method of obtaining values and a grouping mode in a route fixed on this data structure. When we change the zone or window we invert the order of calculation and we begin to group by the last two hours of the day until we reach again the determinant that makes up the 24 hours of the 24 days. In this way, we try to balance the weights of the frequencies since, if we did not invert the order, we would always be rewarding giving more weight to the last hours of the day. This is because the magnitudes of the means and determinants of their frequencies would be greater since they have a greater matrix range. This method of grouping matrices is repeated in sections 6.2.2 and 6.2.3. However, the method for obtaining the scalars that corresponding to each area is defined by the variations in computed paths explained in these sections are defined in Figure 9.

Using this process, we obtain 22 values for determinants and another 22 values for means generated from each submatrix formed by the proposed path plus the determinant of the complete matrix (i.e. 264 variables). This value of the average and the global determinant of each of the windows, making a total of 300 scalars. The structure of the dataset transformed with this method is 2 variables to identify source and destination, and 300 feature variables to analyze.

6.2.2. Obtaining views of the tensor by calculating the maximum frequency in the variable size submatrices

The second approach proposes obtaining the maximum local frequencies [21]("maxpooling or downsampling") of the submatrices by alternating their order of calculation according to the same schemes applied in subsection 6.2.2. With this method, we obtain 22 maximum values of frequency generated by each submatrix of the window defined by the proposed path in Figure 9. The structure of the dataset transformed with this method has 2 variables to identify source and destination and 136 variables to analyze ("features"). This is a widely used method for image processing because it keeps visual coherence when the level of detail is reduced.

6.2.3. Calculating the sum of the frequencies in the variable size submatrices

In this method we calculate the sum of all of the frequencies for each submatrix or defined zones. Following the order of grouping of submatrices in Figure 9, for each cell of the origin-destination matrix we obtain the 22 values of the frequency sums of each of the matrices according to the paths of the arrows in the figure, which define their grouping order with path inversion. The structure of the dataset transformed with this method is composed by two variables to identify origin and destination and 136 feature variables. By accumulating the frequencies the formed variables also achieve the objective of summarizing the frequency of the areas defined by the route.

6.2.4. Calculation of determinants and averages of fixed size submatrices

The orthogonalization process has generated 24x24 variables per window, containing the frequencies added for each hour of the day in 24-day intervals (we use a form of quadrant grouping). As a consequence, each origin and destination (492x492) is composed of 6 windows of orthogonal frequency matrices of 24x24. Each frequency matrix is divided into 16 quadrants or submatrices where we calculate the determinants in each cell of the origin-destination tensor, following the grouping order of the submatrices shown in Figure 10.

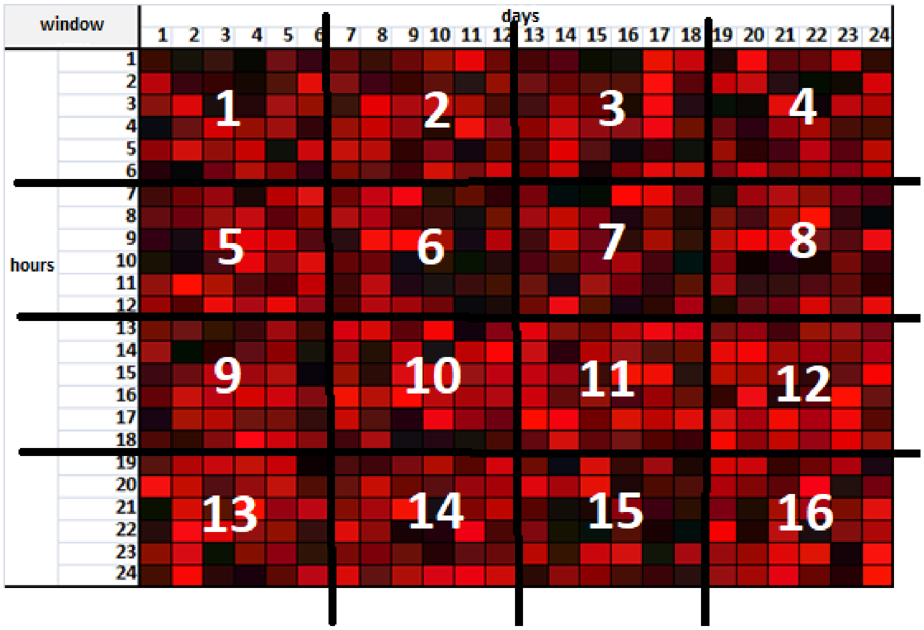


Figure 10: Fixed-window pooling method. Each numbered box (1..16) defines a fixed set of squared submatrices to apply the chosen pooling method. In this figure we show the accumulated frequencies window by hour. This is one of the six windows extracted from the wavelet decomposition previously exposed. The color of the cells indicate the accumulated frequencies for a specific hour.

With this process, we obtain 16 averages and determinants, which are calculated on the frequency of each submatrix of the defined window. The structure of the dataset transformed with this method is 2 variables to identify source and destination and 128 variables to analyze characteristics ("features").

6.2.5. Calculating maximums of fixed-size submatrices

The "Max-Pooling" function [4] is widely used in the application of convolutions in signal processing. We apply it to the selected areas of the frequency tensor in an analogous way We calculate the local maximum value for each of the 16 zones in Figure 10 on each cell of the origin-destination tensor and following the grouping order of submatrices in Figure 7. The result of the process is 16 maximum values of the frequencies generated by each submatrix for each of the 6 defined windows. The structure of the dataset transformed with this method is 2 variables to identify origin and destination and 64 scalar variables as features.

6.2.6. Calculating the sum of the frequencies in the fixed size submatrices

In this method, we calculate the sum of the values of each of the 16 zones in Figure 10. This operation is done on each cell of the origin-destination matrix for each time window considered in Figure 7. With the process, we obtain 16 values with the sums of the frequencies generated by each submatrix for each of the 6 defined windows. The structure of the dataset transformed with this method has 2 variables to identify origin and destination, and 64 candidate scalar variables as characteristics or "features".

6.3. Analysis and validation

This section presents the evaluation of the views generated from the tensor. This is the analysis and validation phase of the classification method, specifically, the Density-Based Spatial Clustering (DBSCAN) and Hierarchical Density-Based Spatial Clustering (HDBSCAN) methods. In order to evaluate the views generated, we executed the classification methods on each of these alternatives and analyzed the quality of the results obtained.

6.3.1. Analysis and validation using DBSCAN

DBSCAN has the following as input parameters: the dataset to be classified, "epsilon" ϵ and "samples". The value of "samples" indicates the number of neighboring points for a point to belong to a cluster, and "eps" reflects the minimum value of the density of the cluster generated for a point to be included in it. The DBSCAN algorithm has been executed on each of the datasets. Each dataset execution is split into 14 sub-executions of the algorithm, with the parameters "eps" and "samples", corresponding to Table 3. This table has been generated in order to limit the parameters of the algorithm, and it has been obtained after a previous sweep with different ranges, in which we have detected convergence in the executions. These combinations of eps and sample ranges are proposed as input parameters for the DBSCAN classification algorithm and are specified in these 14 eps/sample combinations.

The convergence of the DBSCAN algorithm for the proposed parameters has been tested on each of the cases by performing the 14 execution variations shown in Table 3. For each execution of the algorithm, we reflect the corresponding resulting silhouette indexes for the validation of the model are show in the tables obtained. These results can be seen in the results analysis section, for each of

	eps	samples
1	0.0000001	1
2	0.99620491989562243	1
3	0.0000001	2
4	0.000001	1
5	0.00000000001	1
6	0.0000000001	1
7	0.000000001	1
8	0.0000000005	1
9	0.05	1
10	0.5	1
11	0.75	1
12	0.075	1
13	0.005	1
14	0.0000000000001	3

Table 3: Selected parameters for the DBSCAN executions

the views generated from the tensor. The silhouette index has been calculated to measure each classification results to measure the quality of the cluster structures obtained. This is why two values of the silhouette index are obtained. One is for a sample of 1000 classified elements (indicated in the results of the execution as "Silhouette Coef. (1000)") the other is for a sample that is made up of 10000 elements, which makes it more precise (indicated as "Silhouette Coef. (10000)"). In some isolated cases of the range, no results are obtained, and the execution returns a memory error because some of the generated clusters contain a point cloud that has similar dimensions to the size of the input dataset. This is marked in the results tables 5, 8 with an asterisk. Each of the extract methodologies outlined in 6.2 refer to their evaluations according to their indexation within each section heading number.

In the case of Section 6.2.1, the means and determinants are passed on to the classification process by means of variable groupings. The table is shown since all of the values obtained for the table entries in its 14 runs are < 0.1. These values of the silhouette indexes denote a lack of structure in the clusters obtained.

The results obtained for the 14 variations of the proposed parameters for Section 6.2.2 (Maximum frequency in the variable size submatrices) are those shown in Table 4.

Table 4 shows that when using the same parameters, the silhouette coefficient is greatly improved by using variable groupings and the local maximum of each grouping, and we obtain many values above zero. However, the clusters obtained according to the weights in Table 1 indicate a lack of structure.

When evaluating the results of the method proposed in section 6.2.3 with submatrices of variable size, in which the scalar extraction technique is based on performing the sum of the defined areas on defined sliding matrix windows, we

	eps	sampl.	Num.clusters	Silhou.Coef.(1000)	Silhou.Coef.(10000)
1	0.0000001	1	186016	0.096	0,195
2	0.99620491989562243	1	158366	0,032	0,044
3	0.0000001	2	5143	-0,202	-0,32
4	0.000001	1	186016	0,081	0,19
5	0.00000000001	1	186016	0,088	0,192
6	0.0000000001	1	162044	0,039	0,188
7	0.000000001	1	186016	0,088	0,185
8	0.0000000005	1	186016	0,082	0,193
9	0.05	1	1722178	-0,031	-0,019
10	0.5	1	162044	0,039	-0,04
11	0.75	1	160285	0,062	0,019
12	0.075	1	170640	0,023	-0,031
13	0.005	1	186016	0,093	0,19
14	0.0000000000001	3	2410	-0,227	-0,143

Table 4: Results for 6.2.2: obtaining the scalars with local maximum variable groupings

obtain the results shown in Table 5. These results indicate levels of clustering that generate strong structures. In the cases marked with (*), no results were obtained for the proposed density values; the algorithm overflowed the memory because it grouped most points in a single cluster. Option 13 in Table 5 is the one with the highest silhouette coefficient, so it represents a strong cluster structure.

In the validation of the method described in section 6.2.4, in which the characteristics of the matrix are extracted by obtaining scalars with the means and determinants of the fixed quadrants defined in the matrix, we obtain the results shown in Table 6, this table shows that in no case do the associated values exceed 0.4. This indicates that the groupings made are rather casual and do not reflect significant coherence.

The evaluation of section 6.2.5, the max-pooling method on the defined submatrices of fixed size gives us the results shown in Table 7. This results have silhouette index values that denote that this method has obtained reasonable structures (Silhouette Coef. > 0.5).

Finally, Table 8 shows the evaluation of the proposed method in section 6.2.6, which is scalar by summing up fixed-size matrices. In this experiment, the highest silhouette coefficient reaches the value of 0.4, which denotes a weak structure or one that could be artificial.

When analyzing these results as a whole, most of the satisfactory executions made on the obtained load show that "Samples" has a value of 1 (since the experiments that obtain values of the silhouette index above 0.5 are considered satisfactory). Thus, any point of the observations that has at least one neighboring point within the established radius will make both of them the core of density, making the zone grow while there is at least one reachable point. It can be perceived that, with the increase of this value, the coherence of the clusters

	eps	sampl.	Num.clusters	Silhou.Coef.(1000)	Silhou.Coef.(10000)
1	0.0000001	1	36342	0,709	0,793
2	0.99620491989562243	1	(*)	(*)	(*)
3	0.0000001	2	5509	0,634	0,713
4	0.000001	1	36342	0,682	0,795
5	0.00000000001	1	36342	0,722	0,796
6	0.0000000001	1	36342	0,704	0,794
7	0.000000001	1	36342	0,711	0,796
8	0.0000000005	1	36342	0,731	0,795
9	0.05	1	36342	0,731	0,792
10	0.5	1	26980	0,522	0,594
11	0.75	1	(*)	(*)	(*)
12	0.075	1	36342	0,703	0,793
13	0.005	1	36342	0,709	0,797
14	0.0000000000001	3	3212	0,593	0,691

Table 5: Results for 6.2.3: obtaining scalars by means of sums with variable groupings

	eps	sampl.	Num.clusters	Silhou.Coef.(1000)	Silhou.Coef.(10000)
1	0.0000001	1	99289	0,265	$< 0{,}1$
2	0.99620491989562243	1	62536	0,17	$< 0{,}1$
3	0.0000001	2	5589	$< 0{,}1$	$< 0{,}1$
4	0.000001	1	99289	0,259	$< 0{,}1$
5	0.00000000001	1	99289	0,223	0,354
6	0.0000000001	1	99289	0,263	0,357
7	0.000000001	1	99289	0,237	0,358
8	0.0000000005	1	99289	0,239	0,357
9	0.05	1	92770	0,301	0,383
10	0.5	1	86829	0,26	0,335
11	0.75	1	72468	0,155	0,175
12	0.075	1	92092	0,322	0,387
13	0.005	1	99289	0,251	0,356
14	0.0000000000001	3	2988	$< 0{,}1$	$< 0{,}1$

Table 6: Results for 6.2.4: obtaining scalars with means and determinants by quadrants

	eps	sampl.	Num.clusters	Silhou.Coef.(1000)	Silhou.Coef.(10000)
1	0.0000001	1	56314	0,396	0,571
2	0.99620491989562243	1	47125	0,325	0,417
3	0.0000001	2	9131	0,28	0,407
4	0.000001	1	56314	0,406	0,57
5	0.00000000001	1	56314	0,399	0,562
6	0.0000000001	1	56314	0,391	0,577
7	0.000000001	1	56314	0,386	0,575
8	0.0000000005	1	56314	0,377	0,577
9	0.05	1	56314	0,405	0,573
10	0.5	1	56314	0,407	0,579
11	0.75	1	56314	0,403	0,586
12	0.075	1	56314	0,411	0,575
13	0.005	1	56314	0,375	0,569
14	0.0000000000001	3	5290	0,19	0,389

Table 7: Results for 6.2.5: obtaining the scalar with local maximum

	eps	sampl.	Num.clusters	Silhou.Coef.(1000)	Silhou.Coef.(10000)
1	0.0000001	1	102551	0,215	0,406
2	0.99620491989562243	1	(*)	(*)	(*)
3	0.0000001	2	10965	< 0,1	0,101
4	0.000001	1	102551	0,191	0,41
5	0.00000000001	1	102551	0,222	0,4
6	0.0000000001	1	102551	0,216	0,405
7	0.000000001	1	102551	0,218	0,399
8	0.0000000005	1	102551	0,216	0,401
9	0.05	1	56541	< 0,1	< 0,1
10	0.5	1	(*)	(*)	(*)
11	0.75	1	(*)	(*)	(*)
12	0.075	1	52576	0,14	< 0,1
13	0.005	1	91530	0,198	0,262
14	0.0000000000001	3	5951	< 0,1	< 0,1

Table 8: Results for 6.2.6: obtaining the scalar with sum of quadrants

obtained decreases. Bearing this in mind, any point considered as a neighbour can be included in the cluster in a transitional way. The clustering process will be shaped in this way, gradually bringing together points that are very close to each other.

The object of analysis is a set of data that is made up of variables that accumulate the frequencies between origins and destinations in different time ranges and durations. These conditions characterize a discrete, two-dimensional space in which it is quite common to find very close, almost overlapping points, within the clustering space. This is because the distribution of points is made up of clusters that are far apart from each other, but with a great concentration of points in each cluster. Line 13 of Table 5 defines the parameters that have obtained the best silhouette index in the execution of the model for a simulated sample of 10000 elements, with an "Epsilon" $\epsilon \equiv 0.005$ and ”samples” $\equiv 1$. According to the silhouette index validation scale it represents a strong cluster structure. Which indicates clusters of points that are very close. However the clusters are very far from each other.

There are studies that propose methods for determining density parameters for Epsilon on which to start creating the clusters and the number of points. They propose $n \equiv (dim * 2) - 1$, with dim being the dimension of the dataset to be treated [20]. To determine *epsilon* (eps), some authors [6] propose a heuristic method that is based on obtaining the value of the distance to the neighbouring room $(dim * 2)$ or to the third neighbor if the point from which the query of neighbors is made $(dim * 2 - 1)$ [19] is not included. Then, a dotted graph is made, and the first point in the first valley of the graph is selected as the first local minimum. Any point to the left and above is considered be noise. However, in our field of application the number of neighbouring points defines the quality of the groupings obtained and requires very small grouping radii to obtain consistent results.

6.3.2. Analysis and fine validation using HDBSCAN

The DBSCAN method, whose results were analyzed in the above section, makes the parameter search somewhat experimental. To adjust the parameters, we must know the distribution of the points of our dataset within the space that they define However, in order to use DBSCAN method, is necessary to know almost perfectly how the feature variable selected and the distribution of their points in space in terms of the distances between them. Generalist methods for establishing these parameters do not always work [19].

This means that to obtain good results in the clusters, if we use clustering methods that require the number of clusters to be obtained as an input parameter, we must have an expert who initially approximates the number of clusters to be obtained. If, on the other hand, we use DBSCAN, we need an expert to know the differential magnitudes that define the points of the dataset.

After performing the proposed set of executions for the different datasets using the DBSCAN algorithm, we observe that there is a great concentration in the possible groupings found. As these are algorithms that obtain groupings based on densities, there are executions that obtain a single cluster. This indicates that the points of the cluster are very close in terms of the variations of the magnitudes of the frequencies between points. We reach this conclusion because the best results obtained are given by very small epsilon values and a minimum number of points equal to 1.

Once we have detected the best alternative of frequency tensor extract using DBSCAN, we apply the HierarchicalCluster HDBSCAN method [3] to our dataset to refine and improve the results. This classification technique is based on the DBSCAN algorithm, and it varies the values of "Epsilon" ε and integrates the results to find a classification that provides the greatest stability over "Epsilon" ε. This allows having dense clusters but with different densities, unlike DBSCAN, which discards the areas of points with low densities. By using branching and pruning methods for the actual adjustment of parameters of the DBSCAN method, HDBSCAN reduces its computational complexity. It also has the advantage that only the minimum cluster size is required as an input parameter. This reduces the range of possible variations in the two initial parameters of the DBSCAN and its computational complexity, making it possible to program a range of parameter variation for running the executions.

Due to its single input parameter,the use of the HDBSCAN algorithm alows to perform a sweep of runs on the best alternative to obtain the frequency tensor extract. In the previous Section 6.3.1 in which we evaluated the structures obtained for the different proposals of extracts from the tensor, the option of obtaining sums of the variable windows cited at Section 6.2.3 was the one that obtained the strongest structures. This classification technique was applied, giving the values of 2 to 15 elements per cluster in this new validation phase.

A continuous assessment of the silhouette index in the range of minimum cluster sizes between 2 and 14 was carried out to compare the evolution of the local maximums, obtaining the results shown in Table 9.The local maximum of the silhouette index reaches the value of 0.88475073, wich according to Table 1, indicates that the structures obtained after the process are strong. These results correspond to a minimum cluster size of 4, and after the execution of HDBSCAN algorithm we obtained 2302 clusters.

Figure 11 shows how the number of clusters obtained as a function of the parameter follows a logarithmic distribution, while the silhouette index obtained represents variations in a very small range (between 0.87 and 0.88). Within the selected range, the choice of the best parameterization option would be marked by the turning points of the silhouette index and their local maximums in relation to the number of clusters obtained at that point. In our data load, although the best silhouette index is given in the case of “min_cluster_size”$\equiv$ 4, the structure quality value obtained for. “min_cluster_size”$\equiv$12 is similar. In this case, the information is concentrated in 896 routes, as opposed to the 2302 clusters obtained in the other case that optimizes the index.

7. Conclusions

In this work we have proposed a technique that is based on the Wavelet transformation processes to perform a discrete transformation for the extraction

min_cluster_size	silhou.Ind.	n_clusters
2	0,88459775	4406
3	0,88176818	2981
4	0,88475073	2302
5	0,87550227	1902
6	0,88400876	1612
7	0,88311139	1426
8	0,8798205	1275
9	0,87890409	1154
10	0,87829777	1045
11	0,87200316	954
12	0,88433195	896
13	0,88117464	816
14	0,87846965	764

Table 9: HDBSCAN results for minimum cluster sizes in the range of 2 to 14

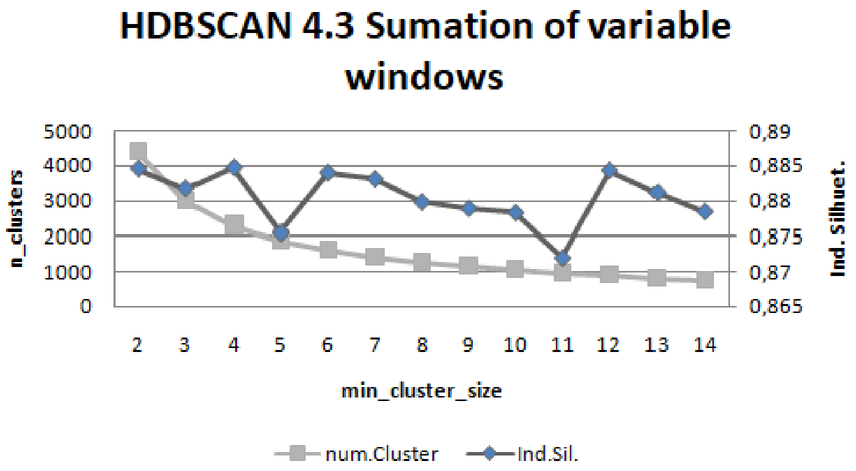


Figure 11: HDBSCAN graphical representation for minimum cluster sizes in the range of 2 to 14.

of patterns that relate time series of points belonging to any geometric space. We have demonstrated that the proposed methodology facilitates the data analysis process by generating a frequency tensor that represents the discrete Wavelet transformation, which, in turn, concentrates all of the temporal information of the movements for use in subsequent treatments. Different methods have been proposed to generate a set of tensor extracts, and the method that has the best resulting structures (defined by the Silhouette Index obtained for each execution of each classification analysis process) has been proposed. The best method of those proposed is based on frequency addition by variable window extraction.

The method of obtaining the frequency tensor and the process of generating the extract proposed in the use case achieved strong structures after analyzing the results of the subsequent classification process.

The conclusion of the case study does not cover the characterization of the cluster that would allow the identification of patterns, to delimit travel habits, means of transport used, and other possible variables related to mobility. This will be addressed in future work.

For the adaptation of the classification methods used and the search for parameters and results, we have presented the following classification analysis method :

- Phase 1 : dimensioning and characterization of sample parameters by means of DBSCAN. This is done once to empirically obtain the ranges of groupings obtained. During this phase, you select the pooling method on the tensor that obtains the best cluster structures is selected.
- Phase 2: parameterization and execution of the classification analysis, using the HDBSCAN method. First, the minimum cluster size that optimizes the results in terms of strong structures is obtained based on silhouette index and the number of clusters obtained. To achieve this, it is executed for the best generated extract evaluated in the previous phase. After executing the proposed range, the optimum structures are obtained, whose classification results are the proposed shift patterns for the initial frequency tensor.

References

[1] Murga M. Alexander L., Jiang S. and Gonzalez M. Origin-destination trips by purpose and time of day inferred from mobile phone data. *Transportation research part C. Emerging Technologies, vol. 58*, pages pp. 240–250, 2015.

[2] Lorenzo G. D. Liu L. Calabrese, F. and C. Ratti. Estimating origin-destination flows using mobile phone location data. *Pervasive Computing, IEEE 2011,10, 4*, pages 36–44, 2011.

[3] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jorg Sander. *Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection.* ACM, 1–51, 2015.

[4] U.; Masci J.; Gambardella L. M.; Schmidhuber J. Ciresan, D. C.; Meier. Flexible, high performance convolutional neural networks for image classification. *International Joint Conference on Artificial Intelligence*, page 1237–1242, 2011.

[5] S. Colak, L.P. Alexander, B.G. Alvim, S.R. Mehndiratta, and M.C. Gonzalez. Analyzing cell phone location data for urban travel: current methods, limitations, and opportunities,transportation research record. *Journal of the Transportation Research Board.2526*, pages 126–135, 2015.

[6] Kriegel H. P. Sander J. Xu X. Ester, M. *A density-based algorithm for discovering clusters in large spatial databases with noise*, pages Vol. 96, №34, pp. 226–231. 1996.

[7] Hidalgo C. A. Gonzalez, M. C. and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature 453, 2008*, pages 779–782, 2008.

[8] Kang Lee J. Senn O. Ratti C. Holleczek T., Yu L. and Jaillet P. Detecting weak public transport connections fromcellphone and public transport data. *Proceedings of the 2014 International Conference on Big Data Science and Computing*, 2014.

[9] G.A. Yang Y. Ferreira Jr J. Frazzoli E. Gonzalez M.C. Jiang, S.Fiore. A review of urban computing for mobile phone traces: current methods, challenges and opportunities. *In Proceedings of the 2nd ACM SIGKDD international workshop on Urban Computing*, page (p. 2). ACM.(2013), 2013.

[10] N.; Bakiras S. Kalnis, P.; Mamoulis. On discovering moving clusters in spatio-temporal data. *In Proceedings of the International Symposium on Spatial and Temporal Databases*, page 364–381, 2005.

[11] Rousseeuw P.J. Kaufman L. Finding groups in data; an introduction to cluster analysis. *Wiley*, 1985.

[12] Gunnar R atsch Koji Tsuda Klaus-Robert Muller, Sebastian Mika and Scholkopf Bernhard. An introduction to kernel-based learning algorithms. 2015.

[13] G. Openheim y J.M. Poggy M.MIsiti, Y Misiti. *Wavelet Toolbox, Users Guide*. The Math Works, Inc. 2000, 2000.

[14] Trasarti R. Monreale A., Pinelly F. Wherenext: a location predictor on trajectory pattern mining. *KDD*, pages 637–645, 2009.

[15] F. Morchen. Time series feature extraction for data mining using dwt and dft. *Technical Report No.33*, 2003.

[16] C. O'Neil. *Weapons of Math Destruction*. Broadway Books, 2016, September.

[17] Longhi D. Gaborieau J. Pronello, C. Smart card data mining to analyze mobility patterns in suburban areas. *Sustainability MDPI*, 2018.

[18] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1986.

[19] Kriegel H. Sander J., Ester M. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery 2,2*, pages 169–194, 1998.

[20] Ester M. Kriegel H. Xu x. Schubert E., Sander J. *Dbscan Revisited, Revisited: Why and How You Should (Still) Use DBSCAN*, page Vol.42 No.3 Article 19. ACM Transactions on Database Systems, 2017.

[21] Mikio Takagi Seishi Takamura. A hibrid lossless compression of still images using markov model and linear prediction. *Image Analiisis and Processing:8th International Conference, ICIAP95*, pages 199–207, 1995.

[22] S.Mallat. *A theory for multiresolution signal decomposition: the wavelet representation*, pages vol. 11, no.7 pp. 674–693. 1989.

[23] Qu Z. Song C. Limits of predictability in human mobility. *Sciencemag.org*, page March, 2010.

[24] Smith K. Hyndman R. Wang, X. Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, pages 13–335–364, 2006.

[25] J. Han Y. Li and J. Yang. Clustering moving objects. *KDD'04, Seattle, WA*, page 617–22, 2004.

[26] Arno Siebes Zbigniew R. Struzik. The haar wavelet transform in the time series similarity paradigm. Septiembre 1999.