A New Tool Based on GIS Technology for Massive Public Transport Data

Nieves R. Brisaboa, Guillermo de Bernardo, Pablo Gutiérrez-Asorey(✉), José R. Paramá, Tirso V. Rodeiro, and Fernando Silva-Coira

Universidade da Coruña, Centro de investigación CITIC, A Coruña, Spain
{nieves.brisaboa,guillermo.debernardo,pablo.gutierrez,jose.parama, tirso.varela.rodeiro,fernando.silva}@udc.es

Abstract. In this work, we present the design of a novel geographic information tool for the analysis of public transportation data.

The widespread integration of user traveler cards have enabled public transport operators to generate and store large amounts of data related to user movements within the transport network. However, these authorities are seldom equipped to efficiently exploit these data in order to produce a comprehensible analysis of the transport network usage. This is not only due to the sheer amount of data in need of processing, but also because most public transport operators only validate the travel card on boarding, whereas data referring to transfers and alightings are generally unavailable.

Thus, the system we propose not only addresses efficient storage and exploitation of big datasets, but also the reconstruction of complete journeys by using a prediction algorithm to deduce the alighting stop for each boarding. Furthermore, we also provide the transport operators with easy-to-use means of visualizing and analyzing the data through a graphical interface.

Keywords: Geographic information systems · Compact data structures · Big data visualization

1 Introduction and Motivation

The main goal of this work is to build a new tool based on geographic information technology for the analysis of trips in the urban and/or metropolitan public transport network (including metro, commuter train, trams, urban buses, and

This work was supported by the CITIC research center funded by Xunta de Galicia, FEDER Galicia 2014-2020 80%, SXU 20% [ED431G 2019/01 (CSI)]; Partially funded by [RTI-2018-098309-B-C32]; MCIN/ AEI/10.13039/501100011033 [PID2020-114635RB-I00], [PID2019-105221RB-C41], [PDC2021-120917-C21], [PDC2021-121239-C31]; by GAIN/Xunta de Galicia [ED431C 2021/53] GRC; by Xunta de Galicia/Igape [IG240.2020.1.185]; by [IN852D 2021/3] CO3, UE, (FEDER), GAIN, Convocatoria Conecta COVID and by Xunta de Galicia [ED481A/2021-183].

P. Fournier-Viger et al. (Eds.): MEDI 2022, CCIS 1751, pp. 121–135, 2022.
https://doi.org/10.1007/978-3-031-23119-3_9

interurban buses). This application will exploit the fact that currently most trips on public transport begin with the validation of a traveler's card. This validation allows recording a large amount of information about the trips and opens the door to derive the transport needs of citizens, including the global use of the network, the most used stops and their peak hours, the occupation of means of transport, etc. Transport authorities and operators are currently facing major problems in adequately storing all the big data generated (millions of records of data produced daily by traveler cards, generating an extensive history of use), so that later they can be analyzed and exploited efficiently. Besides, due to the particular characteristics of many public transport means, complete information on journeys and transfers is sometimes not available since only entry points to the transport network are stored; e.g., buses and many subways where users do not have to validate their card when alighting.

Despite the existence of algorithms capable of deducing the final stops of the travelers just analyzing their accesses to the transport network [1,2], many transport network administrations are still using on-site surveys to create origin-destination matrices in order to analyze the use of the network made by travelers. These surveys have a high cost, so they are only done sporadically (normally once every several years), and therefore they do not reflect the changing reality of people's mobility. For example, they do not reflect the changes caused by the pandemic context, where the use of public transport has changed radically, they are not even adequate to reflect more gradual changes such as those caused by the appearance of new forms of transport such as electric scooters, rental bicycles, etc.

Thus, although the entities in charge of the transport network potentially have very valuable information on the use of the network, in general, they do not have the appropriate tools for its analysis and exploitation, which limits its usefulness.

This work will tackle three main technological shortfalls in the state of the art:

- Lack of efficiency in the storage, integration and management of travel data in relational databases.
- Difficulties in implementing efficient algorithms to accurately infer final stops of travelers.
- Difficulties in visualizing trajectory data with conventional GIS technology.

2 Related Work

Object trajectories has attracted a lot of attention recently [7] as the number of vehicles and devices equipped with GPS technology or other location means has grown explosively. The collected data have many applications, including traffic management, analysis of human movement, tracking animal behavior, security and surveillance, military logistics and combat, and emergency-response planning [5]. The vast quantities of data, however, can make storing, processing, analyzing them a challenge.

In this work, we present a project that will develop a complete system to collect, store, process, and analyse transportation data. This implies that this system will include a wide variety of technologies and methods from data structures and algorithms [4,12] to sophisticated analysis techniques [1,2,6].

3 Previous Concepts and System Architecture

This section includes some basic concepts. First, we need to precisely define some vocabulary that refers to particular concepts about the normal behavior of a public transport network that will be used in the following sections:

- **Station**: we consider a station as the physical location where one or more lines of any means of transport stop.
- **Line**: this refers to a line providing public transit of any means of transport. It is also important to distinguish between line and **route**. A route is an ordered, consecutive succession of stations that defines a path within a line. At the bare minimum, in the dataset we used, every line has two routes corresponding to the same series of stations in two possible directions.
- **Stop**: with this we are referring to the act of a specific means of transport, following a specific route, stopping at a station and allowing for the boarding and alighting of passengers.
- **Trip and trip-stage**: A trip is the complete journey of a user from one point (called origin) to another (called destination). A trip may require boarding multiple means of transport. A trip-stage starts when the user boards any means of transport at a stop, and ends when the user alights at another stop. Therefore, we define a trip as a sequence of one or more coherent trip-stages. The origin of a trip is the boarding stop of its first trip-stage, and the destination is the alighting stop of its last trip-stage.

Having these concepts and the challenges mentioned above in mind, we combine in the same architecture classic and well-established engineering practices with cutting edge technologies.

The proposed system is a classical full-stack solution composed by three main layers: storage, middle communication and user interface. Figure 1 depicts every module encompassed in its corresponding layer, notice how the modules with the most significant contributions are represented with dashed lines. The following is a brief summary of the goal of each layer:

- **Storage layer.** On the one hand, it is responsible for storing basic data in classical databases (considering the spatial dimensions). On the other hand, this layer is in charge of processing the raw data records, calculating final stops for all the trips and storing the information in enhanced structures designed to reduce space usage while preserving good performance. Next sections will delve into these two advanced components.
- **Communication Layer.** Classic middle tier that serves the information saved in the storage layer to the interface layer.

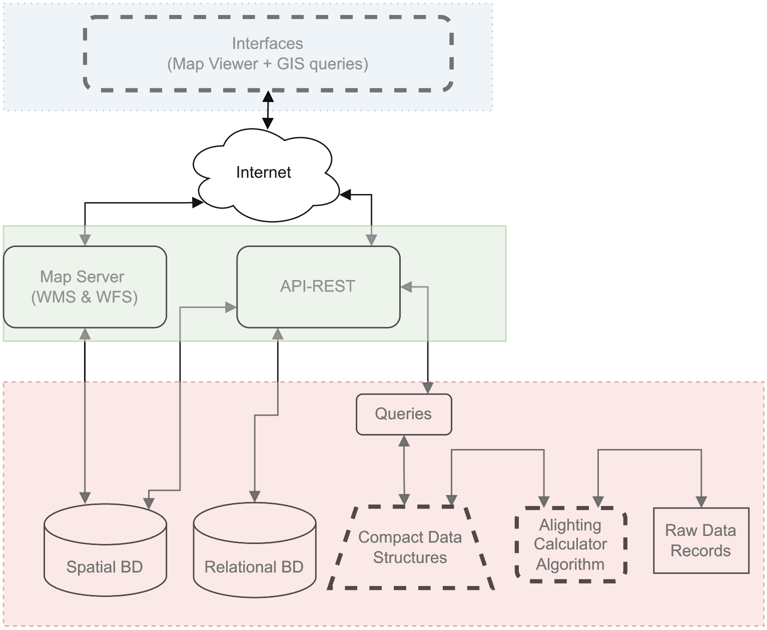


Fig. 1. System architecture with its three distinct parts: storage layer (red), communication layer (green) and user interface layer (blue). (Color figure online)

- **User interface layer.** This layer will provide advanced query interfaces on available public transport travel data. It will provide intuitive interfaces for performing a wide set of predefined but parameterizable advanced queries. For its implementation, we will use standard Web development technologies. Commonly, those results will be presented in the form of maps in the map viewer; in other cases, the results will be presented as lists, statistics, etc. A more detailed description of the contributions on this layer can be found in Sect. 7.

4 Raw Data, Clean Data and Alighting Estimation

All the data used for this project was provided by the Regional Consortium of Transportation of Madrid, Spain.[1] The data encompasses all the transactions of every user card, within any means of public transport, in the city of Madrid, in the years 2019 and 2021.[2] This amounts to a dataset of more than 250 GiB. For reference, just the month of January of 2019 includes 151,094,796 records.

The first major challenge of this work was to interpret, clean and process the raw data. These data consists of twenty-four similarly formatted tables (one for

[1] https://www.crtm.es/.

[2] 2020 was excluded from the analysis due to the anomalous use of the network caused by the pandemic scenario.

each month of the years 2019 and 2021) where each record represents one single transaction on the public transport network of Madrid.

A transaction is tied to an user card id, as well as a specific date and time. Transactions also specify the type of user card, the type of user and the type of discount, if any, that was applied to that transaction. Every transaction is also accompanied by a *paypoint code*, that refers to the specific stop where the transaction took place, as well as a validation code that identifies the conditions in which the transaction took place.

The data include all current user cards in the public transport network of the city of Madrid, including user cards for the elderly people, tourist user cards, etc. We also have data regarding single tickets sold not tied to any user card. However, given that the prime interest of this project is to explore the continuous movements of people, we chose to discard such data (corresponding to a 1.3% of the total number of transactions) for the time being.

We were also provided with a set of tables defining the topology of different public transport networks on Madrid, one for each means of transport included in the transactions dataset, those being: subway, suburban train, trolley car, urban bus, and interurban bus. Each record on these tables describes a specific stop in full detail (including name, station and line that it belongs to, exact geographical coordinates, among others) tied also to a *paypoint code*. By using this value, we are able to correlate each transaction with the full information of the stop where it took place.

Figure 2 shows the complete statistics of transactions by means of transport in the month of January 2019. In summary, 46% of the total number of transactions corresponds to subway stations, making that the most used means of transport by a wide margin, with urban bus (24%) being the second most used, followed by suburban train (16%), interurban bus (14%) and, lastly, trolley car (2%).

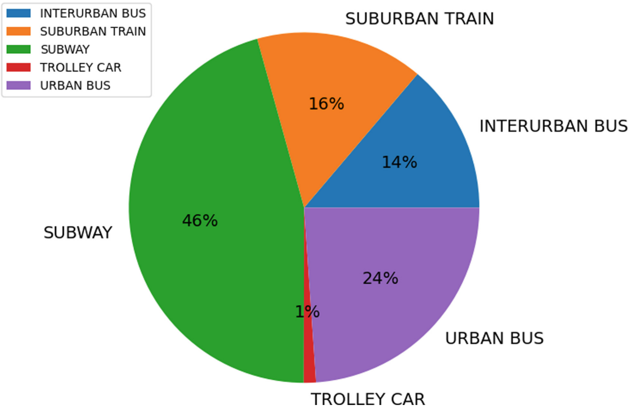


Fig. 2. Transactions per transport means in January 2019, including all types of users.

The distinct validation codes could be used to classify every transaction into three broad categories as follows:

- **Boarding**: the user card was validated when boarding at a specific stop.
- **Alighting**: the user card was validated when alighting at a specific stop.
- **Change of vehicle**: the user card was validated when performing a mandatory change of vehicle in some special stations.

It is important to note that for most means of transport, there is no standardized process for validating the user's card upon alighting, meaning that the vast majority of the available data relates only to boardings without a subsequent alighting. 88% of the total number of transactions are classified as boarding transactions, with 8% being classified as alightings and 4% as changes of vehicle.

Table 1 shows the percentages of every transaction type for every different means of transport. Observe how, for both subway train and trolley car, more than 40% of the transactions are labeled as alightings. This is because for those means of transport users are expected to validate their card upon alighting on most stations. In contrast, on the subway network, only a very small number of stations pertaining to special subway segments that are closed off the regular subway network, demand the users validate their cards on alighting, thus only a 1.64% of the transactions on subway are labeled as such. Finally, as there is never a validation on buses when alighting, there are no transactions labeled as alightings on those means of transport.

Table 1. Percentage of transaction types for each different means of transport

Means of transport	Boardings	Alightings	Changes of train
Subway	90.03%	1.64%	8.33%
Suburban train	57.02%	42.98%	0%
Trolley car	57.23%	42.77%	0%
Urban bus	99.99%	0%	0.01%
Interurban bus	100%	0%	0%

Given that a trip is a concatenation of trip-stages consisting of both a boarding and an alighting, and the state of the data we have just described, it is not possible for most of the real trips to be derived from the data we were provided with. This presents a serious problem for us, given that the ultimate objective of this project is the construction of trips based on the data of boardings and alightings for every trip-stage. For this reason, we concluded it necessary to develop some kind of method in order to estimate the possible alighting stop for every boarding registered in our dataset.

5 Alighting Estimation

In this section, we propose an algorithm based on simple rules with the aim of predicting possible alighting stops for every trip-stage based only on boarding data. Figure 3 shows the general flow of that algorithm.

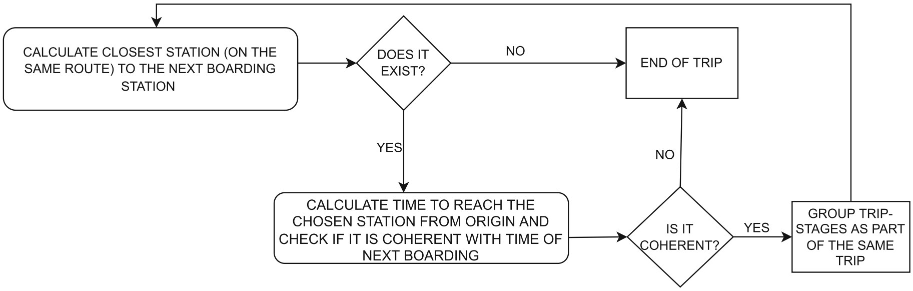

CALCULATE CLOSEST STATION (ON THE SAME ROUTE) TO THE NEXT BOARDING STATION
DOES IT EXIST?
NO
END OF TRIP
YES
NO
CALCULATE TIME TO REACH THE CHOSEN STATION FROM ORIGIN AND CHECK IF IT IS COHERENT WITH TIME OF NEXT BOARDING
IS IT COHERENT?
YES
GROUP TRIP-STAGES AS PART OF THE SAME TRIP

Fig. 3. Flow of the algorithm for the prediction of alighting stop and grouping trip-stages into trips.

In essence, the algorithm consists of two simple steps.

Step 1): For each pair of consecutive boardings, we want to calculate a possible alighting for the first boarding. For this, we search for the closest stop to the second boarding accessible from the first. In other words, it is assumed that there is at least some amount of continuity in the user's movements, so we will start by establishing the possible alighting of a boarding in a stop close to the immediate next detected boarding.

If there is no stop accessible from the first boarding sufficiently close to the second boarding, we determine that it is not possible to deduce an alighting stop at this step, and also that the trip-stage defined by the first boarding and subsequent alighting must constitute an end-of-trip.

Step 2): Assuming Step 1 was successful in identifying a possible alighting stop, it is now necessary to check whether it is reasonable to assume that this created trip-stage can be linked with the next trip-stage as part of a longer trip. In order to accomplish this, we compare the time of the first boarding plus the estimated time needed to arrive at the predicted alighting stop, to the time of the second boarding.

If the two trip-stages are really part of the same trip, the user needs to have been capable of arriving to the estimated destination of the first trip-stage on time to start their second trip-stage without an extended delay, meaning that if there is a difference between the two times larger than a given threshold, it is not reasonable to consider the two trip-stages as part of the same trip, as most likely the user has spent some time on their first destination before starting a new trip.

We apply these rules to every pair of consecutive trip-stages of the same user card. This results in a collection of trips where every trip-stage that is not the last trip-stage of a trip includes a candidate alighting stop.

Of course, it is necessary to define an appropriate threshold for each of these steps. For the first step we need to define the maximum distance beyond which we consider a stop is no longer close enough to another stop for it to be chosen as a valid destination of the previous trip-stage.

Conversely, we need to define the maximum time delay allowed between the estimated time of arrival at a destination and the next boarding for the two trip-stages to be linked as part of the same trip. Note how this method is based on the same continuity principle used in [1]. These results were improved by applying machine learning techniques in [2].

One obvious shortcoming of these simple rules is that we cannot calculate a candidate alighting stop for any trip-stage that is also the last trip-stage of a trip. For this, we need to make yet another assumption about the movements of users in transport, that is, that there is a degree of symmetry in a user's journeys between different days. For example, if a user always starts her day by boarding at a specific stop that we could assume is located near her home, and her last boarding of the day is on an stop that has a possible destination close to that first boarding of the day, it would be reasonable to assume that such destination could be the last alighting stop of the day for this user, and that this last trip of the day represents the act of "going back home".

By searching for regular patterns in the movements of the users and applying this principle of symmetry, it should be possible to further complete the data of the trips.

Furthermore, given that we were provided with information about user's profiles and the types of user cards used for every transaction, we are currently investigating possible alternatives to incorporate this data in the flow of the algorithm in order to enhance the prediction.

To close up this section, note that, while our dataset is certainly lacking in real data about alightings, we can still perform a validation of the prediction algorithm we designed by comparing the alighting stops suggested by the algorithm on trip-stages for which we do have both the boarding and alighting data (this would be true for most-trip stages on suburban train, trolley car, as well as certain segments of subway network).

6 Trip Representation: Aggregated and Disaggregated Data

To face the Big Data 4V, parallelism is the usual choice. However, the basic data structures of most NoSQL systems are not very suitable for the analytical needs of this project, since those systems rely on parallelism with simple data structures like key-value pairs. Instead, our target is to use more complex data structures such as a trajectory data warehouse [9]. Data warehouse is the natural choice for data analysis since they provide grouping capabilities, which includes the possibility of using of pre-computed aggregated data for speeding up queries.

The classical systems based on relational databases or native data warehouse systems need very expensive dedicated hardware to cope with the amount of data that needs to be handled in this project. Therefore, we will use a new alternative based on compact data structures [8], which are capable of storing data in compressed form but, unlike classical compression methods, they allow to extract (decompress) portions of the data without having to decompress the whole dataset. This opens the door to a new computation paradigm, where data reside in main memory all the time in compressed form avoiding disk accesses, thus yielding faster access times [10]. In addition, taking advantage of the savings in space consumption, compact data structures are usually equipped with indexes (also compressed) or precomputed aggregated data, which helps even more to obtain faster query times. In recent years, compact data structures for managing aggregated data have already been designed for multidimensional data [3].

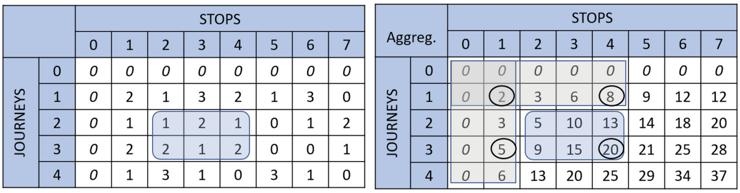
We are already developing a novel data structure for this project inspired by previous compact data structures. One of the most relevant structures in this context is the Topology & Trip-aware Compact Trip Representation (TTCTR) [4], a compact data structure that stores the individual trips of each traveler. Each trip of a traveler is represented as an sorted set of tuples $<s_i, l_i>$, where s_i is the identifier of the stop and l_i the line to which belong. A vocabulary stores the unique identifier that is assigned to each tuple and trips are represented using this vocabulary. TTCTR concatenates and sorts the set of trips and stores it in a modification of the well-known compact and self-indexed data structure Compressed Suffix-Arrays (CSA) [11].

In the experiments, a dataset of 10 millions of trips occupying 165,39 Mb was reduced to 38.16 MB, that is, only 38.16% of the original space.

As this structure works only for individual trips, the authors also proposed T-Matrices (Trip Matrices) [4]. They emerged as an application of image rendering techniques to the particular study of public transport loads. The main idea is to simplify the approach to store aggregated data while speeding up cumulative queries. In the public transport domain, three queries of this type stand out:

- Aggregations by stop (e.g. number of boardings on a specific vehicle at a given stop S).
- Aggregations by time-interval (e.g. number of boardings at any stop of the line on 08/01/2022).
- Aggregations by stop and time-interval (e.g. number of boardings at a given stop S on 08/01/2022).

The simplest way to store the loads of a transport mean would be a classic table, being one axis the actual stops and the other one each journey of a given vehicle, as it is depicted in Fig. 4 (left). However, to solve any of the queries listed above, it would be necessary to iterate through each of the desired cells. That cumbersome process could be avoided by storing the data in an aggregated matrix, summing all values from the top-left corner to the bottom-right corner. Thus, we can compute the total sum of any submatrix in O(1) time



		STOPS							
		0	1	2	3	4	5	6	7
JOURNEYS	0	0	0	0	0	0	0	0	0
	1	0	2	1	3	2	1	3	0
	2	0	1	1	2	1	0	1	2
	3	0	2	2	1	2	0	0	1
	4	0	1	3	1	0	3	1	0

Aggreg.		STOPS							
		0	1	2	3	4	5	6	7
JOURNEYS	0	0	0	0	0	0	0	0	0
	1	0	2	3	6	8	9	12	12
	2	0	3	5	10	13	14	18	20
	3	0	5	9	15	20	21	25	28
	4	0	6	13	20	25	29	34	37

Fig. 4. T-matrices. The left table represents the actual gathered values while table on the right represent the same information using an aggregated approach. (Color figure online)

using the formula: $\mathsf{sumSubMat}((\mathsf{x}_1, \mathsf{y}_1), (\mathsf{x}_2, \mathsf{y}_2)) \leftarrow M(x_2, y_2) - M(x_2, y_1 - 1) - M(x_1 - 1, y_2) + M(x_1 - 1, y_1 - 1)$.

In the example of Fig. 4, each cell represents the amount of travelers that boarded into a given vehicle during one of its journeys at a given stop. In order to calculate how many travelers boarded on journeys 2 and 3 from stops 2, 3 and 4 (submatrix highlighted in blue) it would be necessary to perform $1 + 2 + 1 + 2 + 1 + 2 = 9$. Yet, using the aggregated solution (right), the same result can be achieved using the formula *sumSubMat* only accessing four cells: $20-8-5+2 = 9$.

One interesting query not tackled in that work is to precalculate the total number of users who used a particular line and their distribution throughout the day. This distribution indicates the percentage of use of a line every hour, in intervals of 15 min or another range of time. For example, a given line X is used on average by 250 people, where 15% boarded between 9:15 and 9:30, 10% boarded between 9:30 and 9:45 and so on. This allows us to know, among other information, the peak hours of the stops and lines, the real use of the lines, population movement patterns, etc.

The structures previously described in this section are not able to resolve this type of queries efficiently. For this reason, we propose a new method based on the creation of a vocabulary that allows us to solve them. The method is based on the fact that the distributions of use will be repeated very frequently between different lines. For instance, many trips in the night hours will have very low or no usage, mornings will have peak usage due to people going to work, etc.

The main idea is to create a vocabulary with the different percentages of distribution that occur in the transportation area during a day. Then, each trip is assigned the number of passengers and also the identifier in the vocabulary of its distribution of use.

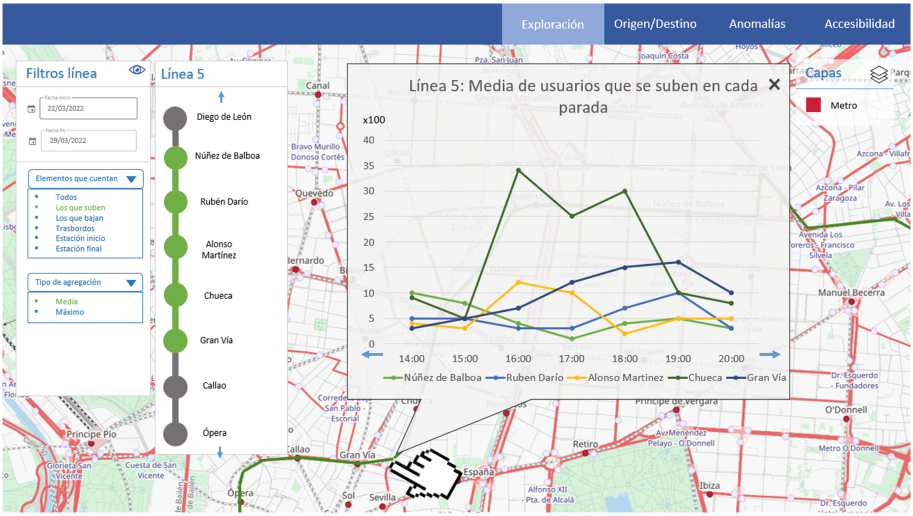
7 User Interfaces for Analysis

The systems exploits the trip information using four main user interfaces, that are devoted to different dimensions of the transport management:

- Detailed network usage: this involves analyzing how many passengers have used a given line, or boarded at a given stop, during a specific day or range of days, and how this number of passengers evolves during the day.

- Origin-destination demand: this involves displaying the overall demand by users for specific origin-destination trips (i.e. number of full trips, possibly including several trip-stages and multiple transport means, that start at a given stop and end at a given stop in a given range of dates).
- Anomaly analysis: this involves identifying potential origin-destination pairs that have sufficiently high demand but do not have a convenient combination of lines to do the trip in reasonable time. This kind of analysis is mainly useful for tactical and strategical decisions related to transport planning.
- Accessibility analysis: this involves identifying the connectivity of specific stops. The goal of this analysis is to be able to determine, from any given stop in the transport network, the average time required to reach any other point of the network.

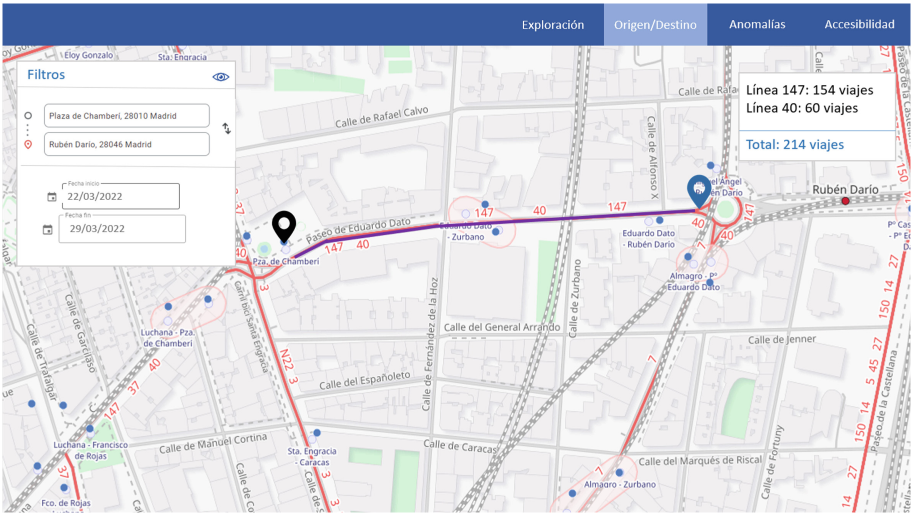
Figure 5 displays a sample interface for detailed network usage analysis. This is the most detailed query interface, and also answers the most basic usage analysis queries: The interface displays all the lines for all transportation means, and users can select a specific line (or, more specifically, a route of that line) in order to obtained detailed information about the number of passengers using that specific route. For a given route, user can filter data to a given temporal range, and to a subset of stops. The system will compute the desired metric for each selected stop in the line and display the evolution of that metric in predefined temporal intervals (in the example, 1 h). The metrics available for each stop include the number of passengers boarding, alighting, switching transportation means, as well as starting or ending trips in the given stop. This information can be obtained from aggregated data stored using the data structures introduced in Sect. 6.


Fecha inicio: 22/03/2022

Fecha fin: 29/03/2022

Fig. 5. Network usage analysis interface.

Figure 6 displays the origin-destination analysis interface. This interface provides a higher level analysis of the transportation habits of passengers, as it focuses on characterizing full trips of users. The goal of this interface is to provide an intuitive way to identify the demand on the network between two given points.



Plaza de Chamberí, 28010 Madrid

Rubén Darío, 28046 Madrid

Fecha inicio 22/03/2022

Fecha fin 29/03/2022

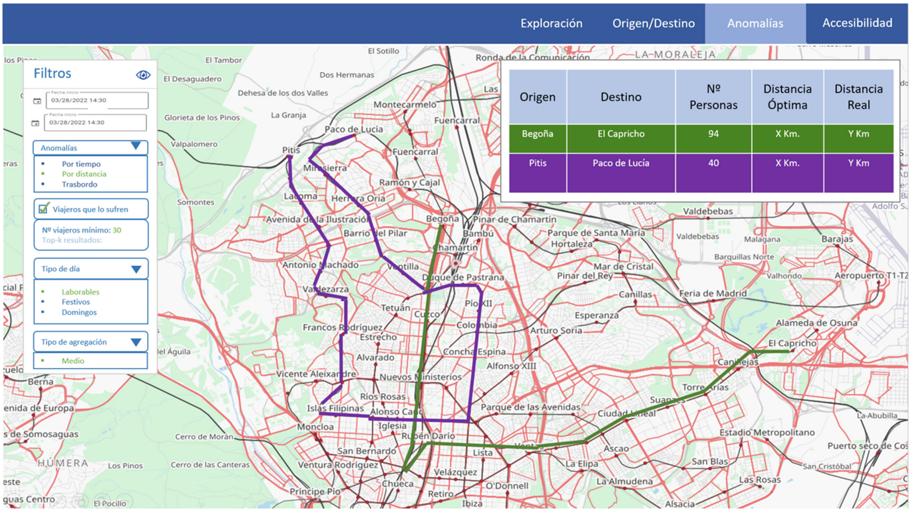
Línea 147: 154 viajes
Línea 40: 60 viajes

Total: 214 viajes

Fig. 6. Origin-destination query interface.

The query interface, as displayed in Fig. 6, displays a map with all the stops in the region, where the user may select any two stops as origin and destination, to check the demand between them. Additionally, the results can be constrained to a given range of dates in order to analyze detailed behavior for specific dates or events. The interface retrieves an aggregation of all the user trips corresponding to the given origin-destination, partitioned by line, and displays all of them in the map. Each different route for the given origin-destination is displayed at the same time, and summarized information related to the number of trips using the corresponding route is shown to the user.

Figure 7 displays the user interface for anomaly analysis. Unlike the previous interfaces, this one does not require the selection of specific stops or lines, since its goal is to automatically detect all potential anomalies. The user may select a range of dates (typically, to include specific events, or to discard older data if changes in the network have occurred), as well as specific conditions on the dates (e.g.: weekends only), the minimum demand (i.e. the number of passengers that perform the corresponding origin-destination trip) and the anomaly metric. The relevant anomaly metrics are 3: time anomaly (the trip takes much longer than it would according to the distance), distance anomaly (the trip is much longer than it could be), and vehicle changes (the trip requires changing lines



Fecha inicio 03/28/2022 14:30

Fecha inicio 03/28/2022 14:30

Anomalías
- Por tiempo
- Por distancia
- Trasbordo

☑ Viajeros que lo sufren

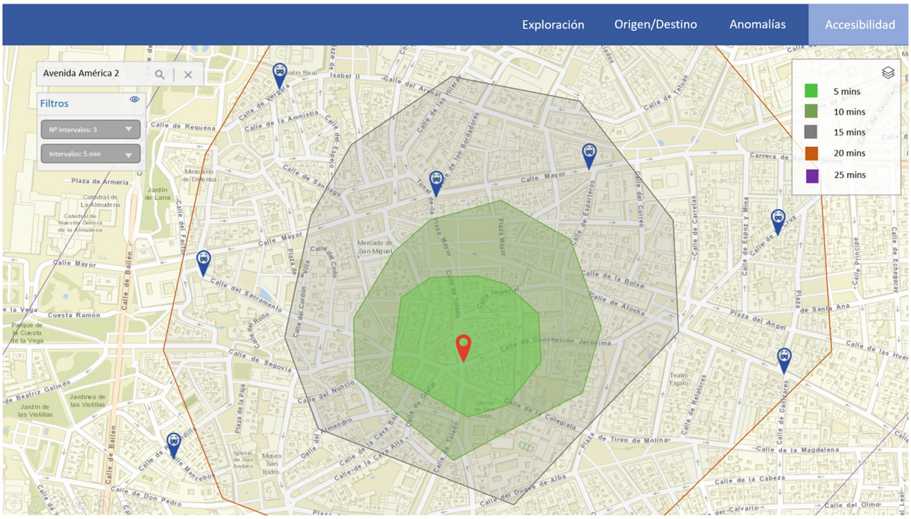
Nº viajeros mínimo: 30
Top-k resultados:

Tipo de día
- Laborables
- Festivos
- Domingos

Tipo de agregación
- Medio

Origen	Destino	Nº Personas	Distancia Óptima	Distancia Real
Begoña	El Capricho	94	X Km.	Y Km
Pitis	Paco de Lucía	40	X Km.	Y Km

Fig. 7. Anomaly analysis interface.



Avenida América 2

Nº intervalos: 3

Intervalos: 5 min

Fig. 8. Accesibility analysis interface.

too many times). Given these parameters, the system computes and displays all the origin-destination pairs that contain an anomaly.

The accessibility analysis interface, displayed in Fig. 8, provides a simple interface to select any stop in the transport network and display the estimated time from the given stop to any other stop in the network. This is displayed in the form of an isochrone map, that isolates the different regions that can be reached in fixed time intervals. Notice that, since different stops may have very different connectivity and transfer times, the interface also provides the ability to manually select the number of isochrone lines generated and the time interval used to construct them.

8 Conclusions

In this work, we have presented an ambitious project under development of a novel geographic information system for the analysis of public transportation data. Several challenges force us to adopt new state of the art techniques.

First, the amount of data being produced at a very high rate daily in large cities and their metropolitan areas. In our case, we include in the proposed system compact data structures to build a completely new trajectory data warehouse. The reason of this choice is that classical data warehouses, such as those based in relational model, first, are not adequate to build a trajectory data warehouse and second, they have serious problems to deal with very large volumes of data. The alternative of NoSQL systems is also not suitable since to obtain aggregated data, all must be done on the fly.

Second, the systems designed so far have only basic analysis queries, and thus, we designed a new set of queries and query interfaces that provide very complex data analysis for a transportation analyst.

Acknowledgments. We wish to acknowledge the cession of the data and support of the *Consorcio Regional de Transportes de Madrid* (https://www.crtm.es/), especially Juan Elices Torrente and Luís Criado Fernández.

References

1. Alsger, A., Assemi, B., Mesbah, M., Ferreira, L.: Validating and improving public transport origin-destination estimation algorithm using smart card fare data. Transp. Res. Part C Emerg. Technol. **68**, 490–506 (2016)
2. Assemi, B., Alsger, A., Moghaddam, M., Hickman, M., Mesbah, M.: Improving alighting stop inference accuracy in the trip chaining method using neural networks. Public Transp. **12**(1), 89–121 (2019). https://doi.org/10.1007/s12469-019-00218-9
3. Brisaboa, N.R., Cerdeira-Pena, A., López-López, N., Navarro, G., Penabad, M.R., Silva-Coira, F.: Efficient representation of multidimensional data over hierarchical domains. In: Inenaga, S., Sadakane, K., Sakai, T. (eds.) SPIRE 2016. LNCS, vol. 9954, pp. 191–203. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46049-9_19
4. Brisaboa, N.R., Fariña, A., Galaktionov, D., Rodeiro, T.V., Rodríguez, M.A.: Improved structures to solve aggregated queries for trips over public transportation networks. Inf. Sci. **584**, 752–783 (2022)
5. Gudmundsson, J., Laube, P., Wolle, T.: Movement patterns in spatio-temporal data. Encyclopedia of GIS **726**, 732 (2008)
6. Kopczewska, K.: Spatial machine learning: new opportunities for regional science. Ann. Reg. Sci. **68**, 713–755 (2021). https://doi.org/10.1007/s00168-021-01101-x

7. Mahmood, A.R., Punni, S., Aref, W.G.: Spatio-temporal access methods: a survey (2010–2017). GeoInformatica **23**(1), 1–36 (2018). https://doi.org/10.1007/s10707-018-0329-2
8. Navarro, G.: Compact Data Structures: A Practical Approach. Cambridge University Press, USA (2016)
9. Pelekis, N., et al.: Towards trajectory data warehouses. In: Giannotti, F., Pedreschi, D. (eds) Mobility, Data Mining and Privacy. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-75177-9_8
10. Plattner, H., Zeier, A.: In-Memory Data Management: Technology and Applications. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29575-1
11. Sadakane, K.: New text indexing functionalities of the compressed suffix arrays. J. Algorithms **48**(2), 294–313 (2003)
12. Zheng, Y., Zhou, X. (eds.): Computing with Spatial Trajectories. Springer, New York (2011)