v. 4, n. 5, 2024

Journal of Engineering Research

MODELING THE DATABASE ARCHITECTURE FOR PUBLIC TRANSPORTATION BIG DATA WITH A FOCUS ON DATA INTEGRATION-ETL

Kaio Gefferson de Almeida Mesquita
http://lattes.cnpq.br/4288305631665524

Kleberson Leandro da Rocha
linkedin.com/in/rochakleberson

Gabriel Amorim Rabelo Nobre
http://lattes.cnpq.br/8786853451934733

Abstract: The process of analyzing and planning public transport systems has some aspects that must be treated as an integrated part, from understanding the phenomenon to analyzing the data that represents it. There is a gap in the work that proposes to use massive transport data, regarding the inadequate treatment of the data or lack of structuring to store it. The objective of this work is to present a method for structuring a public transport database, using transformation, mining and natural language processing techniques. The method is divided into: Contextualization; Cleaning and transformation; and Loading and evaluation. The results demonstrate that data produced by humans presents more inconsistency than data generated by machines. The GPS and Ticketing bases, in addition to being integrated, achieved a compatibility and treatment rate above the usual average of 65%.

INTRODUCTION

A transport system can be defined as a set of elements and their interactions between the demand for travel and the supply that must satisfy them (Cascetta, 2009). Therefore, the process of analysis and planning of public transport (TP) systems has some aspects that must be treated as an integrated part, from understanding the phenomenon studied to the way it is represented, whether by analyzing travel patterns, factors that influence these patterns or even the method of reconstruction of trips (ANTP, 1997).

Ortúzar and Willumsem (2011) discuss the role of transport planning, where the system's supply must be consistent with demand, and, therefore, understanding the pattern of users' movements and how this demand varies over time, can help in proposing supply planning (Mesquita et al., 2017). A superficial understanding of this system mainly influences decision makers' inability to identify the population's travel patterns (Agard et al., 2006). Traditionally, travel patterns are identified in Origin/Destination (O/D) Surveys, the most common tools for obtaining information about the mobility of a city, carried out in the field, with the population of the study area, through sampling of this population. The disadvantages of such surveys are that they are expensive to conduct, time-consuming and often do not cover more than 3% of the population. One of the advantages is the possibility of obtaining the individual's socioeconomic data. This type of study is generally carried out in most Brazilian capitals every 10 years (Guerra et al., 2014). In addition to these databases, other complementary data can be used to make the information more robust and closer to reality during inference of unknown parameters and trip reconstruction. As an example, there is the base of user registrations, line shapes, zoning shapes and *smartphone* location data (WU et. al., 2018).

In the late 1990s, smart card payment systems were incorporated in some cities, such as Washington D.C. and Tokyo, also known as Automated Fare Collection – AFC (Electronic Ticketing System - SBE), allowing fare payment (for example: travel validation) through Smart Cards and reading equipment installed in vehicles (Zhao et al., 2007; Munizaga and Palma, 2012). In addition to SBE, many cities around the world have also been adopting Automatic Vehicle Location (AVL) systems, composed of Global Positioning System (GPS), for real-time location of vehicles, having a logistical aspect, but which is currently being used to understand the variability in the supply and demand of public transport (Zhao, 2004).

In addition to fare collection, smart cards also continuously collect passenger behavior. This way, the size of the data can become so large that it can surpass the processing capacity

of conventional means, adding to the volume, the speed of extraction, the variety of data and the value of the information, composing the so-called Public Transport Big Data (BD-TP) (Han et al., 2011; Kurauchi; Schmocker, 2016). Therefore, it is essential to clean Big Data for inconsistencies due to human error (swiping the card on the validator more than once) and equipment error (not identifying latitude and longitude due to interference from tunnels and buildings). The explosion of data generated contributes to the challenge of how to store and manage it. Therefore, new forms of analysis and tools are needed that match the characteristics of the data.

The objective of this work is to present a database structuring method focusing on massive public transport data, from data extraction, processing and evaluating the relational model of Fortaleza's public transport system. The central problem refers to the manipulation of massive data inappropriately, interfering with the analysis of prediction and characterization of a transport phenomenon. Therefore, this proposed structure enables characterization analysis, travel reconstruction and demand modeling, as it allows high performance of data queries with low computational cost, after implementation. In addition to presenting the problem, the work consists of 6 more stages: Review of massive data modeling; Integration method – Extraction, transformation and loading; Extraction and identification of data types; Cleaning and transforming variables; Final database structure and evaluation; and Final considerations.

DATA MODELING

In the mid-2000s, some authors believed that the growth of open sources (OSINT – Open Source Intelligence) would facilitate the work of the bodies responsible for the activity and would bring a significant cost reduction. At the time, it was already clear that the flood of data generated by the "democratization of information" and the popularization of communication technologies would increase the burden on decision-makers. An important aspect to highlight is the dizzying growth in the amount of information available, a phenomenon that implies information overload for the analyst and the decision-maker.

MASSIVE DATA MANAGEMENT

The most diverse types of data are generated daily, which do not always have a structure. Making it necessary to know the format of this data to obtain the best way to process and store it (Isotani; Bittencourt, 2015). There are several elements involved in generating data, from personal computers running information systems, cell phones with applications, to even the most diverse types of sensors and capture tools. The fact is that there is a great diversity of data, storing the most diverse information. In this second point, data is distributed around the world according to its format and storage structure, which, in general, are classified as structured, semi-structured and unstructured, with only 20% of the total data produced worldwide in format structured.

Structured data are those that are stored in a previously defined structure, traditionally software uses them in the form of Relational Database Management Systems (SGBDR). The main characteristic of SGBDRs is that they were built to guarantee the integrity of the stored data, their entire structure is based on tables built based on relationships. Semi-structured data is data that has a structure, that is, a meaning can be recognized, however, unlike structured data, this structure is not previously defined. This structure is incremental and changes over time. Examples of semi-structured data are: Excel

spreadsheets, CSV files, XML documents, JSON documents. In turn, unstructured data is the most commonly present in information systems. It is not possible to obtain a structure from these, and to extract knowledge about such data it is necessary to carry out pre-processing. Examples of unstructured data are: texts, images, sound files, videos and the most diverse multimedia files. Due to the rise of the concept of the internet of things and sensing, data can be collected in real time (streaming data) or in pre-defined batches (batch data) (Mello, 2000).

In June 1970, researcher Edgar Frank changed the history of databases by presenting the relational model in the article entitled "A Relational Model of Data for Large Shared Data Banks", in which he discussed a proposal for storing data, which would be stored in tables that must be related (Codd, 1970). This way, with a large amount of data extraction, the basis of what is currently known as Big Data began to form. This is a term that has as its closest literal translation "large amounts of data" and is also the term established by the International Business Machines Corporation (IBM) to determine the large amount of data generated by information systems. However, Big Data is related to the more data entities can collect, the more decision-making power they can obtain. In its main definition, Big Data is known as a scenario that contains the sum of volume, speed and variety (3 Vs), which, when together, generate an information value. Volume is the central factor, speed refers to information reaching the decision maker's hands in the fastest time and variety is related to both the devices that collect data (e.g. information systems, GPS, video cameras, IoT devices, etc.), regarding the structure of such data.

There is great difficulty in defining how large a data set must be to be considered a volume of data in Big Data. The first aspect to be taken into consideration is the integration of volume, speed and variety, as well as the definition given by Dumbill (2012) in his work that evaluated the processing capacity of various database architectures, where he states that Big Data is data that exceeds the processing capacity of conventional database systems, in which the volume of data is very high and requires fast processing, which is not provided by traditional database architectures and to gain value From this data, it is necessary to choose an alternative way to process it. Carillo (2018) points out Big Data trends for the next decade such as the Growth of IoT (Internet of Things); More accessible artificial intelligence; More present predictive analysis; Growth of Quantum Computing and Smarter Cybersecurity.

STRUCTURES: NOSQL X SGBDR

In addition to learning techniques, it is necessary to have a suitable place to store the estimated data, which can be primarily in relational or not only relational environments. Database Management Systems, or Relational Database Management Systems, are, simply put, software that is responsible for managing access to data. A database is made up of 4 components: The Database, DBMS, Exploration Language and additional programs. The DBMS is just one part of this set and is responsible for interfacing data with applications and users, encapsulating them, ensuring their security and integrity. SGBDRs are characterized by relationships between tables, which implement relational models. Among the main RDBMSs on the market, SQL Server and Oracle stand out among the paid ones, while among the free ones there are MySQL and PostgreSQL. In data scenarios, a current component is the: Not Only Structured Query Language (NoSQL) databases, which emerged in the 2000s, following the rise of large companies such as Amazon and Google,

which increasingly produced data gradually, having the need for scalability in writing and especially reading data after the increase in cloud-oriented technologies. In general, the main advantage of using NoSQL databases is the use of vertical data partitioning, unlike relational databases, which do so horizontally. This means that in NoSQL databases, data can be distributed independently without that it is necessary to send an entire set to a particular node or disk.

Discovering association rules in relational databases is one of the data mining tasks that has the greatest number of practical applications. Data Mining emerged in the early 1990s, from the gathering of ideas from different areas such as artificial intelligence, databases, statistics, and data visualization. The main motivation for the emergence of data mining is the fact that organizations have been continuously storing a huge amount of data about their businesses in recent decades. Due to their wide applicability, association rules are among one of the most important types of knowledge, which can be mined in databases (Han et al., 2011; Kurauchi; Schmocker, 2016). These rules represent relationship patterns between items in a database. As previously presented, there may be several types of variables in a database, each linked to an information field. Association rules allow connections between fields (from different tables) and are made possible by primary and foreign keys. Still dealing with relational modeling, this has among many objectives to store data ensuring the highest possible level of integrity. The main strategy for this is called normalization. Data normalization is the first step towards achieving success with an intact data model, since if these standards are respected, redundancies and inconsistencies can be avoided. Kimball and Ross (2011) state that there are already at least 10 normal forms (NF), all originating from the first three: (I) 1NF: the first normal form deals with the atomicity of attributes, prohibiting compound, multi-valued attributes and nested relationships. (II) 2NF: the second normal form is related to the functional dependence on the primary key. To be in second normal form, the table must be in first normal form and none of the fields that are not keys may not depend on the primary key. (III) 3NF: the third normal form is related to the so-called transitive dependence, that is, a field must not depend on another "non-key" field. To remove transitive dependency, you must identify fields that are transitive dependents on other fields and remove them.

ETL – EXTRACT, TRANSFORM AND LOAD

In general, it is quite difficult to define which step is most important in determining the database architecture, but a strong candidate is the ETL (Extract, Transform and Load) step. Briefly, this is the step responsible for removing the data from the source, preparing it and storing it in a database, in order to integrate bases from different sources and unify them. When it is stated that this is the most important step in the entire process, it is precisely because this step consumes around 80% of the time of a data analysis implementation project (Nagabhushana, 2006).

Although it is a consolidated method in data science, there are no transport studies that focus on this level of data processing, before proposing any exploratory analysis, even knowing the gains in improving analysis accuracy with this process.

The extraction process is responsible for capturing data from sources, as a type of recovery of only necessary data, starting from one source. Such data sources can be the actual tables or simply copies that have been loaded. An important point of extraction is

the diversity of data sources, which can be text documents, XML, JSON, CSV, or several integrated sources. This step is generally delimited with a collection periodicity, for example data extraction every 30 min, or real-time extraction. Moving on to the second stage, there is the one that requires the greatest computational effort among the three, the transformation stage. As a first step, it is necessary to define the data integration, transforming the fields into a single standard and making them ready to be stored. This step seeks to transform the raw data as it is collected according to the architecture and typing specified in the project. It is at this stage that pre-processing is carried out, in which duplicate data, integration, value replacement, field cleaning and any other necessary transformations are identified. Once the previous steps leave the data ready, already collected and transformed, the load step is responsible for storing the data in the database. On load, the DBMS is fed with new data, so that the multidimensional database tables are updated to contain the new data. It is worth noting that there is another approach, analogous to this, called ELT (Extract, Load and Transform), in which the extracted data is stored in its raw form and only a set necessary for a query is transformed.

STRUCTURING, TREATMENT AND STORAGE METHOD

As the first step of the data integration method (relational data integration), it is necessary to know the data well and store it in a structure that allows easy access and high performance. Therefore, this data must be properly treated, respecting the grain size of its types and the normalization of the relational database. Figure 1 represents the method corresponding to data processing and structuring the database, divided into 3 stages, and developed using Python, SQL and R languages:

The first stage of the data processing and database construction method consists of obtaining the data. All data (except the dictionaries and terminal shapes that were formulated by the author) were extracted/ granted by the Fortaleza Urban Transport Company (ETUFOR), responsible for controlling, regulating and supervising the public transport system. The extraction of the main bases took place in batch data. The SBE and GPS data correspond to the month of November 2018. This year was chosen because it was pre-pandemic, and because it contained sufficient data from all databases.

The data processing process, known as cleaning and transformation, can be divided into the treatment of missing values and the treatment of extreme values. It is very unlikely that the data will be organized and clean. Missing values can harm models. It could be a problem with the data source or the extraction process. It is necessary to visualize whether there is a pattern of missing values, in this case verified by Data Mining. Another way to deal with the problem is to simply remove the line in which the value appears or pairs of values, or try to fill in this data. Database queries and joins were created for this level of verification. The case of treating extreme values occurs when they deviate from the average values. An observation that is very far from the data standard. They were checked for univariate and multivariate classes of data. They can be caused by generalized data entry, errors in experiments, intentional or natural observation of the data set. You can simply remove them or treat them individually. Both treatments were performed at this stage. Continuing, the specific treatment for each type of data was discussed, after separating into groupings and defining the types of variables. The definition of data types and corrections to field titles, as well as their standardization,

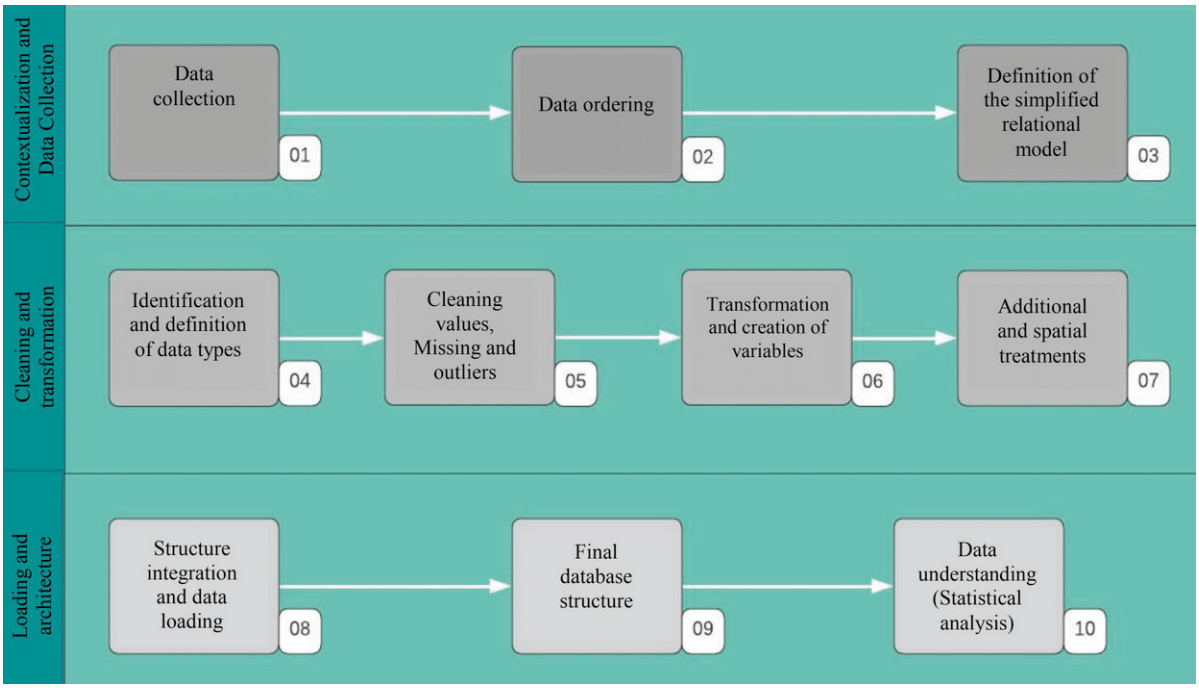

Figure 1: Database Structuring Method for Public Transport Big Data

guided data storage and queries respectively. Therefore, a Natural Language Processing (NPL) script was developed to correct and make line naming compatible. Still in this topic, all available data was ordered and grouped to build relational tables. Having defined the fields and types of each variable, it was possible to build a preliminary relational model that served to show the degree of relationship and dependence of the tables, this degree of dependence that guided the order of data processing, as some data sets cannot be handled if your dependencies are not defined and ready for use. With the prior structuring defined, the cleaning was organized, and an important point at this stage is that some jobs transform the data before cleaning, that is, null data, with incorrect formatting and outliers, are being transformed, wasting time processing that could be invested in other stages.

With the data properly cleaned and transformed, the final relational model was built, which comprises the integration of all the bases raised with some degree of relationship, and which will enable complex queries at a quick and intuitive level of this data. Therefore, the data was loaded into the Relational Database Management System (SGBDR) using the Python and SQL languages, that is, from this moment on, several csv, xml, shp files are no longer necessary, as they are all in a single secure structure that is easily accessible to researchers. The appropriate flowcharts and scripts that exemplify how the process actually occurred are also presented at all stages. At the end, a summary was presented with indicators that reflect the effort expended in the process. The processing rate was calculated, as the ratio between the final records and the raw records, in addition to the file compression rate, based on the number of records, data size and execution time. The final product of this work enables numerous exploratory analyzes and consultations of public transport data in a quick and easy way, enabling future work to analyze spatio-temporal patterns and reconstruct the travel chain.

COLLECTION, IDENTIFICATION AND RELATIONAL MODEL

The data used constitutes 20 databases, not necessarily 20 files, but categories of information, and these databases are divided into 4 collection groups, from which it was possible to obtain the data. The groups were determined according to the means of data availability. The GTFS data, although made available by the managing body of Public Transport in Fortaleza, is passed on weekly to Google, in a specific format and is therefore placed in a separate group, consisting of 10 files: Fare Attributes; Fare Rules; Routes; Agency; Trips; Travel shape; Calendar; Calendar Date; Stops and Stops Schedule. In Figure 2, there is the data made available by PASFOR, referring to collections of Boarding at the terminals, Transshipment at the terminals, Zoning and Zoning Coordinates, in addition to field research data that were integrated in subsequent stages of user registration. The data extracted from ETUFOR, and which guide the development of the management structure, are: Ticketing, GPS, Dictionary and User Registration. Finally, there is a category of shapes that were made available from other research (Braga, 2019), corresponding to the shape of neighborhoods in Fortaleza and the shape of the integration terminals. In the case of terminals, to allow spatial integration of the bases, since it is information that is repeated in other files. Figure 2 presents the proposal for relational integration and structuring of the various Public Transport databases, which can be replicated for other management bodies.

After this explanation of the relational model, the relational priority was defined, that is, the order in which the data must be processed and consequently, the same order must be maintained for creating the tables and loading them into the database, since there is a degree of dependency between tables causing debugging errors if the ordering is not respected. An important point of this work is that the dictionary file only corresponds to the integration of the vehicle identifier in the GPS data with the car prefix in the Ticketing data. Therefore, an integration of the Ticketing, Dictionary and GPS databases was used. This way, the respective values of the car prefix were identified in the dictionary and the value of the vehicle identifier from the GPS in the ticketing database was saved in the same record. The dictionary file only corresponds to the integration of the vehicle identifier in the GPS data with the car prefix in the ticketing data.

CLEANING AND TRANSFORMATION OF VARIABLES

As previously presented, some databases contained more than one file, which is a fundamental principle of machine learning: processing 1 50MB file has greater performance than 50 1MB files. Therefore, the final product of this stage was a single file for each database, with the exception of ticketing and GPS data, which, although data from one month was used for performance analysis, had a file for each month processed and loaded into the database. First, all data, regardless of the relationship hierarchy, was cleaned and only then transformed. This ensures that erroneous data will not be processed and will take up space in memory for processing, optimizing the task. The most common corrections during this process were incorrect data in columns to which they did not belong, null data, negative travel times, non-existent stops, dates inconsistent with the standard format adopted (YYYY/MM/DD) and numeric values saved as strings. In the case of dates, 4 variations of this same information were found within the database.

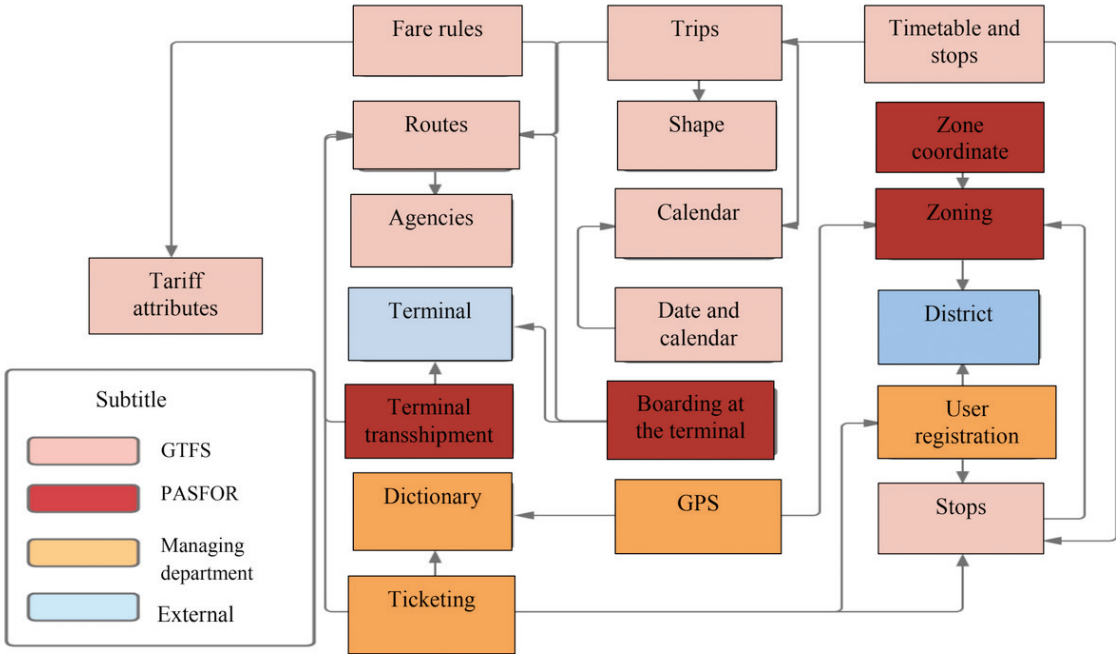

Figure 2: Relational Model for Fortaleza Public Transport Big Data

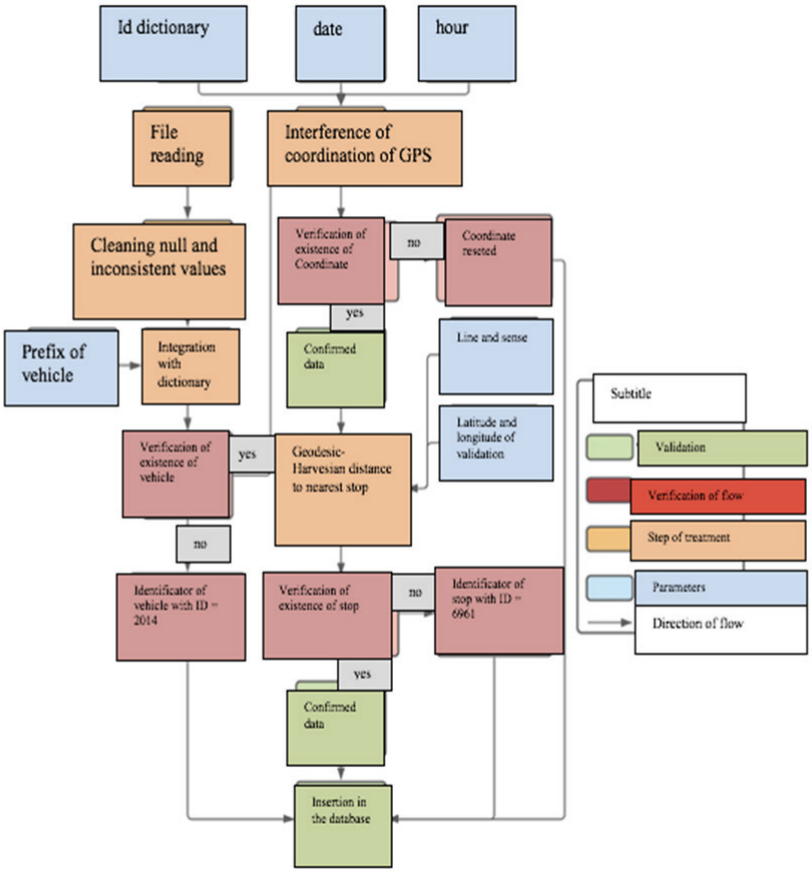

Figure 3: Ticketing data processing method

The stops data is from the GTFS stops file for the year 2018, containing 4969 lines and 12 columns, with much of this data being missing. The processing of this data was simple, as all stops correctly had the latitude and longitude coordinates, requiring only the exclusion of missing columns and being in accordance with the structure defined for the database. The Fare, Agency, Calendar, Fare Rules, and Calendar Date Attribute files are all indirectly linked to the bus lines.

In the user registration data, there was first a file for each month of 2018, all of which were grouped into a single file using the Python language, with just over 400 thousand records. Analyzing the base file, there were many inconsistencies such as: (i) Different number of columns in each file; (ii) Information from a column in the wrong field (Ex.: Telephone number in the neighborhood field); (iii) Names of neighborhoods and cities written in more than one form. The names of neighborhoods were all placed in the same format as the neighborhood table loaded into the database, for this it was necessary to connect to the database and make the necessary queries, using PT-NLP (Public Transport – Natural Language Process), Natural Language Processing algorithm created by the authors to correct and identify Public Transport strings. The initial registration base contained 111MB and the final base loaded with 50MB, with a processing rate of 47.60%.

Continuing, the data from the dictionary was processed, the only problem being finding the data consistent with current affairs, as the files did not contain date identification and for the same vehicle prefix identifier they contained more than one GPS code, depending on the file consulted, with 6 files available with varying sizes. To overcome this problem, ticketing data from a working day from Braga (2019) was used to validate the process.

GPS data was the second database that required the greatest computational power and processing time, with an estimated 90 hours of processing, spaced over two weeks. Each file from a single day contains on average more than 300MB. The processing of the raw GPS data, although it took a lot of time, consisted of cleaning the direction data, formatting the latitude and longitude to decimal type with 6 places, breaking the metric timestamp field into date and time. Each GPS record contains an identifier, with 120 million raw records at the end of the 30 days in November. After processing the GPS, we moved on to processing the last database, ticketing. However, although the ticketing data had smaller files than the GPS data, its direct queries in the preliminary database made the process slower and more time-consuming. These days were processed on 3 machines processing in parallel, taking a total of approximately 336 hours (14 days) uninterrupted to complete the process. The entire ticketing processing process is exemplified in figure 3.

The start is by reading the file and treating null values that are inconsistent with the formatting, including the aforementioned date, time and line name formatting. Subsequently, a query was made in the database to check the existence of the vehicle prefix code and its bank identifier was returned, and if it did not exist, an identifier from the standard dictionary was assigned for non-existent vehicles and automatically saved in the bank. If it is possible to identify the vehicle, this code is used as a parameter along with the date and time, where a new slice (data cut query) is performed with the limitation of just one return to the nearest time taken as a parameter. This way, if no record is returned, the coordinates are assigned null values and automatically saved in the database, cutting off the flow. If so, the coordinates found are stored in new variables,

being a geographic estimate of the validation location given the temporal variation of the GPS. The coordinates, line number and direction of travel are set as parameters for a new query, in this case against GTFS data, returning the coordinates of each stop in the direction of travel sequence, for a single trip in the schedule. Subsequently, the distance from validation to the nearest stop was calculated according to the line and direction, taken from GTFS data. If no stop record was found, a default null stop identifier value was saved, to guarantee the integrity of the bank. In case it is found successfully, all registry data has been stored.

LOADING AND FINAL DATABASE STRUCTURE

Finally, present the loading stage and the final structure of the database, which is one of the products of this work. After the entire cleaning and transformation process of each database, they were loaded into the database using an insert command in SQL. For each of the 20 databases treated and loaded, the records and file sizes were collected in their raw format, after treatment and after loading.

GTFS shape data and user registration data fall into this category, in which it was necessary to create new columns for loading, but user registration data

According to the compression rate, they are 54.43% more compact, due to the number of records eliminated, as it is a difficult database to format, while shape data increased by 58.76%. All databases, except for user registration, had more than 80% of the data processed and recovered for use, while the latter only obtained 47.60%, largely due to the inconsistency of the database with addresses and identification numbers of empty cards. Viewing the compression rate, it is possible to identify that 6 out of the 20 bases had size optimization, but the real gain was in the integrity and security of the data, since the database is encrypted and requires a username and password to access the records. GPS and Ticketing data had a processing rate of 80.35% and 91.62%, respectively, but this value in addition to being an excellent treatment rate (usually 35% of data is discarded due to inadequate data processing), ensuring a high degree of confidence for the next stage of the work (analysis of spatio-temporal patterns), it also makes it possible to validate the treatment method, respecting one of the issues most raised in this work, which is the quality of the data used, with 95.59% treatment rate for all bases and a standard deviation of 7.16%. Finally, as a final product of this work, in addition to the method and relational structure, the data duly treated and stored are presented, which will guide all queries in the next stages of the public transport data mining process to reconstruct the travel chain, as well as any system demand and supply analyses.

ID	Data base	Amount of raw data		Size (KB)	Amount of data when loading		Size (KB)	Treatment fee	Compression ratio
		Lines	Columns		Lines	Columns			
1	Neighborhood	119	5	489	119	4	8	100,00%	-98,36%
2	Zoning	253	5	579	253	4	10	100,00%	-98,27%
3	Coordinates and Zoning	253	*	623	20716	4	804	8188,14%	29,05%
4	Stops	4969	12	312	4696	5	343	100,00%	9,94%
5	User registration	475408	20	111777	226297	20	50940	47,60%	-54,43%
6	Attributes and tariff	2	6	1	2	4	2	100,00%	100,00%
7	Agency	2	7	1	2	6	2	100,00%	100,00%
8	Routes	319	9	21	318	4	15	99,69%	-28,57%
9	Fare Rules	319	5	4	319	3	7	100,00%	75,00%
10	Terminal	9	5	6	9	5	3	100,00%	-50,00%
11	Boarding Terminal	5359	14	474	5221	9	387	97,42%	-18,35%
12	Transshipment Terminal	88942	16	7290	88158	11	7328	99,12%	0,52%
13	Calendar	4	10	1	4	11	3	100,00%	200,00%
14	Shapes	111723	5	4113	108324	6	6530	96,96	58,76%
15	Date and calendar	18	3	1	18	4	3	100,00%	200,00%
16	Trips	82615	9	3308	82586	6	4338	99,96%	31,14%
17	Stop Time	2665108	8	116201	2508282	6	133057	94,12%	14,51%
18	Dictionary	2051	2	24	2014	3	43	98,20%	79,17%
19	GPS	12594520	9	733000	100678235	7	7909000	80,35%	7,9%
20	Ticketing	20739540	9	1750000	19001200	12	2180000	91,62%	24,57%
21	Average	-	-	-	-	-	-	95,59%	29,13%
	Variance	-	-	-	-	-	-	1,67%	66,90%
	Standard deviation	-	-	-	-	-	-	7,23%	61,10%

Figure 4: Summary of database processing results

FINAL CONSIDERATIONS

As explained, the objective of the work

was to present a database structuring method focusing on massive data and processing processes. Among the main updated techniques for this specific data set are natural language processing for recognizing similar words in different databases, optimizing data storage and data mining for integration between GPS and ticketing databases.

The results demonstrate that data produced by human beings, such as user registration, has a higher rate of errors, such as non-existent addresses and card identifiers. The GPS and Ticketing data obtained treatment rates of 80.35% and 91.62% above the average treatment rate of 65% of the few studies that were concerned with this stage. For future work, a comparative performance analysis between the SGBDR and a non-structured bank is recommended. The system presented is the basis for future analysis of travel patterns and reconstruction of Fortaleza's travel chain.

REFERENCES

Agard, B., Morency, C. and Trépanier, M. 2006. Mining public transport user behaviour from smart card data. In 12th IFAC Symposium on Information Control Problems in Manufacturing- INCOM, pp. 17-19.

ANTP: ASSOCIAÇÃO NACIONAL DE TRANSPORTES PÚBLICOS. Transporte Humano: Cidades com Qualidade de Vida. São Paulo, 1997, 312 p.

Braga, C. K. V. (2019) Big data de transporte público na análise da variabilidade de indicadores da acessibilidade às oportunidades de trabalho e educação. 108 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Universidade Federal do Ceará, Fortaleza, 2019.

Cascetta, E. (2009) Transportation Systems Analysis: Models and Applications (2nd ed.). Springer, New York, NY, USA.

Carrilo, D. 10 Big Data Trends You Should Know. 2018. Disponível em: <https://www.kdnuggets.com/2018/09/10-big-data-trends.html>. Acesso em: 14 jan. 2022.

Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6),377–387. doi:10.1145/362384.362685.

Dumbill, Edd. What is big data? An introduction to the big data. 2012. Disponível em: <http://radar.oreilly.com/2012/01/what-is-big-data.html> Acesso em: 22 marc. 2022.

Guerra, A. L.; BARBOSA, H. M.; OLIVEIRA, L. K. Estimativa de Matriz Origem/Destino Utilizando Dados do Sistema de Bilhetagem Eletrônica: Proposta Metodológica. Transportes, [s.l.], v 22, n. 3, p. 26-38, 2014

Han, Jiawei; PEI, Jian; KAMBER, Micheline. Data mining: concepts and techniques. Elsevier, 2011.

Isotani, Seiji; BITTENCOURT, Ig Ibert. Dados abertos conectados: em busca da Web do conhecimento. Novatec Editora, 2015.

Kurauchi, F.; Schmocker, J. D. (2016) Public transport planning with smartcard data. 2016.

Kimball, R.; Ross, M. The data warehouse toolkit: the complete guide to dimensional modeling. John Wiley & Sons, 2011.

Mello, R. dos S. et al. Dados semiestruturados. XV Simpósio Brasileiro de Banco de Dados, 2000.

Mesquita, H. C.; Amaral, M. J.; Carvalho, W.L; Matriz O/D com Base nos Dados do Sistema de Bilhetagem Eletrônica. Congresso Nacional de Pesquisa em Transportes - ANPET, Recife, 2017.

Munizaga, M.A. and Palma, C. 2012. Estimation of disaggregate multimodal public transport origin–destination matrix from passive smart card data from Santiago, Chile. Transportation Research Part C: Emerging Technologies, 24, pp. 9-18.

Nagabhushana, S. Data Warehousing, OLAP and Data Mining. New Delhi, Índia: New Age International, 2006. Ortúzar, J. D.; WILLUMSEN, L. G. Modelling Transport. 4th Edition ed. West Sussex, UK: Wiley, 2011.

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transportation Research Part C: Emerging Technologies 96, 321–346.

Zhao, J. (2004) The Planning and Analysis Implications of Automated Data Collection System: Rail Transit OD Matrix Inference and Path Choice Modeling Examples. Thesis. Massachusetts Institute of Technology, Boston.

Zhao (2007), J.; RAHBEE, A.; WILSON, N. H. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. Computer-Aided Civil and Infrastructure Engineering, v. 22, n. 5, p. 376–387, 2007. ISSN 10939687.