EUBRA BIGSEA
EUROPE - BRAZIL
COLLABORATION OF BIG DATA
SCIENTIFIC RESEARCH THROUGH
CLOUD-CENTRIC APPLICATIONS

D7.3 - Toolbox for GES^3 Data Initial Release

Author(s)	Nazareno Andrade (UFCG), Tarciso Oliveira (UFCG), Leandro Balby (UFCG), Wagner Meira Junior (UFMG), Walter do Santos (UFMG)
Status	Draft
Version	V1.0
Date	11/01/2017

Dissemination Level

☒ PU: Public
□ PP: Restricted to other programme participants (including the Commission)
□ RE: Restricted to a group specified by the consortium (including the Commission)
□ CO: Confidential, only for members of the consortium (including the Commission)

Abstract: Europe - Brazil Collaboration of BIG Data Scientific Research through Cloud-Centric Applications (EUBra-BIGSEA) is a medium-scale research project funded by the European Commission under the Cooperation Programme, and the Ministry of Science and Technology (MCT) of Brazil in the frame of the third European-Brazilian coordinated call. The document has been produced with the co-funding of the European Commission and the MCT.
The purpose of this report is to describe the software and data that implement the various data integration techniques from the Massively connected society - Smart cities, use case. The data and software pointed out in this document constitute the main content of deliverable D7.2, of type "OTHER".

EUBra-BIGSEA is funded by the European Commission under the Cooperation Programme, Horizon 2020 grant agreement No 690116.
Este projeto é resultante da 3a Chamada Coordenada BR-UE em Tecnologias da Informação e Comunicação (TIC), anunciada pelo Ministério de Ciência, Tecnologia e Inovação (MCTI)

Document identifier: EUBRA BIGSEA –WP7-D7.23	
Deliverable lead	Nazareno Andrade (UFCG)
Related work package	WP7
Author(s)	Nazareno Andrade (UFCG), Tarciso Oliveira (UFCG), Leandro Balby (UFCG), Wagner Meira Junior (UFMG), Walter do Santos (UFMG)
Contributor(s)	Fernando Carvalho (UFMG), Átila Martinhs (UFMG)
Due date	31/12/2016
Actual submission date	17/01/2017
Reviewed by	Sandro Fiore (CMCC), Nádia Kozievitch (UTFPR)
Approved by	PMB
Start date of Project	01/01/2016
Duration	24 months
Keywords	Descriptive models, machine learning, smart cities, toolbox

Versioning and contribution history

Version	Date	Authors	Notes
0.1	28/11/16	Walter dos Santos Filho (UFMG)	TOC
0.2	23/12/16	Walter dos Santos Filho (UFMG)	People Paths description
0.3	26/12/16	Walter dos Santos Filho (UFMG)	Description of applications
0.4	27/12/16	Leandro Balby (UFCG)	Review
0.5	29/12/16	Fernando Carvalho (UFMG)	ST-DBSCAN model description
0.6	10/01/17	Walter dos Santos Filho (UFMG)	Applying reviewers' comments
0.7	10/01/17	Tarciso Oliveira (UFCG)	Applying reviewers' comments
1.0	11/01/17	Walter dos Santos Filho (UFMG), Wagner Meira Junior (UFMG)	Final version elaboration

Disclaimer*:* The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the document is believed to be accurate, the author(s) or any other participant in the EUBra-BIGSEA Consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the EUBra-BIGSEA Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the EUBra-BIGSEA Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

EUBra-BIGSEA project aims at developing a set of cloud services empowering big data analytics to ease the development of massive data processing applications. EUBra-BIGSEA will develop models, predictive and reactive cloud infrastructure QoS techniques, efficient and scalable big data operators and a privacy and quality analysis framework, exposed to several programming environments. EUBra-BIGSEA aims at covering general requirements of multiple application areas, although it will showcase in the treatment of massive connected society information, and particularly in traffic recommendation.

The project starts with the analysis of the use case scenarios that will be used for demonstration, but considering those requirements in a broader way. EUBra-BIGSEA is an API-centric project whose main objective is to create a sustainable international (European and Brazilian) cooperation activity in the area of cloud services for big data analytics. In particular, T7.2 aims at improving efficiency and throughput of data scientists and data curators.

The Acquisition and Engineering of Georeferenced Environmental, Stationary, Streaming and Social (GES^3) data (Task 7.2) is related to the Use Case 1 - (UC1) - Data Acquisition (D7.1). In particular, these data come from sources that are related to urban traffic and cover four main data types: stationary data, dynamic spatial data, environmental data, and social network data. Despite that the EUBra-BIGSEA pilot has been initially planned for the data of the city of Curitiba, where the pilot case is being constructed, the EUBra-BIGSEA framework will be applicable to some extent to other scenarios.

Task 7.3 deals with the creation of descriptive models from the GES^3 aforementioned data sources, in order to understand the dynamics of traffic and transportation public services in Brazilian cities. Models applies a specific set of data mining and machine learning unsupervised techniques *clustering, association rules, feature extraction* and common used *summarization* and *aggregation* of data.

Results from Task 7.3 are the first ones using the GES^3 data and computation intensive algorithms. Thus, implemented code has been used as a *proof-of-concept* in diferents WPs to, for example, evaluate infrastructure (WP4), expressiveness of programming abstractions (WP5), identification of security and privacy concerns (WP6) and in the realization of the use cases (WP7). Next steps include integration (indirectly) with resource allocation and evaluation of workload (WP3) and improvements in the implementation (convert pending prototypes to WP4 and WP5 technologies).

Together with Task 7.4, Task 7.3 will provide the *toolbox* needed to implement the complex analytics scenarios of *Routes for People* Use Case (Task 7.5).

1 INTRODUCTION

1.1 Scope of the Document

This document describes the elaboration of a toolbox containing *descriptive models*, their implementation, deployment and application on the smart cities context. It also represents the realization of milestone M7.3

1.2 Target Audience

The document is mainly intended for internal use, although it is publicly released.

At internal level, WP7 members will find in the document some descriptive models that will be applied in use cases described in D7.1. Technical developers from WP3, WP4, WP5, and WP6 will find the scenarios they have to address in their developments. The project success will be also measured in the degree of fulfilment of such issues.

At external level, Data Scientists (referred here as developers of data-analysis-intensive applications) could evaluate whether the issues and solutions addressed are similar to those they have, considering the possibility of adopting the technology. Municipality stakeholders may assess and understand issues inherent to a particular scenario, as well as and possible solutions. Application developers could also understand the kind of problems that could be addressed using the EUBra-BIGSEA components.

The information of this document will be periodically updated in the internal wiki, and the final outcome of the implementation of such requirements will be analysed in deliverable D7.6 (Validation of requirements), by the end of the project.

1.3 Structure

The document has 5 sections, being the first this introduction. Section 2 describes descriptive models and their lifecycle.. Section 3 describes the models constructed and instantiated in the project. Section 4 presents the conclusions.

2 DESCRIPTIVE MODELS

2.1 What are descriptive models

Descriptive models describe an existing subject or set of subjects, being the resulting model derived from observations of the system or scenario being described [R1]. Descriptive models are unsupervised learning functions and do not predict a target value, but focus more on the intrinsic structure, relations, interconnectedness, etc [R2]. Different techniques are available in order to build descriptive models: *clustering, association rules, feature extraction* and common used *summarization* and *aggregation* of data.

A fundamental abstraction of the proposed descriptive models are trajectories, that is, the path traversed by each end user while using public transportation. Trajectories comprise not only dynamic spatial data, but also the other types of data that enrich the trajectory information. Notice that building the trajectories is a challenge by itself, since matching the various types of data to a specific end user trajectory may be very tricky and demand advanced and complex techniques.

2.2 Descriptive models applied in the smart cities context

Descriptive models are the core answer of several smart cities problems and challenges. They are necessary because of the variety and diversity of scenarios to be understood and achieving both summarized and expressive models for the complex phenomena that arise in city-related issues is hard by nature. Researching not only the applications of such models, but also new models that provide superior descriptive patterns. One particular issue that is also addressed by EUBra-BIGSEA is regarding the scalability of these models and how they may be efficiently implemented in cloud platforms.

2.3 Model construction

As mentioned, descriptive models aim to find inherent patterns in a set of data that represent a set of objects or events. Each descriptive model instantiates a pattern, for instance, a group of similar objects or relevante correlations among them. The pattern is characterized by a set of one or more properties, such as entity similarity or the same attributes. These properties are quantified through measures that allow us to rank the patterns that compose a model. Descriptive model construction usually means traversing the search space of the patterns while maximizing some criteria that express, for instance, pattern cohesion. Popular construction strategies are combinatorial, probabilistic, algebraic and based on graphs.

2.4 Model usage

Once built, a model may be used for different purposes. First of all, it may be used as a resource to understand the modeled scenario and its characteristics. Second, it may be helpful as an analysis tool, helping to understand the characteristics of unknown entities that have to be analyzed. It may also be the basis of the construction of predictive models, which may leverage from the properties found in the data. In summary, descriptive models increase our understanding of a given scenario, its entities and their relationships.

3 Toolbox

This section presents a toolbox of models, algorithms and techniques for processing GES^3 data. It is formed by a set of models, sample applications and respective documentation, aiming to be used by EUBra-BIGSEA partners as processing and data analytic tools.

3.1 People Paths

Socio-spatio-temporal analysis of people movement in a city from bus ticketing data.

Source code and documentation with setup instructions available at https://github.com/analytics-ufcg/people-paths.

3.1.1 Description

People Paths is an application which performs a descriptive analysis on bus GPS and passenger ticketing data, finding paths taken by Public Transportation city users in a time period, and matching the paths origin/destination locations to city area social data: population, income and literacy rate. Such information is not easily available, because the passenger ticketing data do not include the GPS coordinates (this information is only available in bus GPS data). The Census data are used to aggregate trips according to a set of economical, social and educational characteristics. In order to join the first two data sets with Census data, we used shapefiles with boundaries of regions matching Census data.

Tests were realized using data from the cities of Campina Grande and Curitiba. Description and availability of data are listed in Table 1.

Data source	Description	Data policy
Bus GPS Data	Buses GPS record for a given time period.	Samples available only to EUBra-BIGSEA partners. Data owner has plans to open these data in 2017.
Ticketing data	Passenger ticketing records for a given time period, with some card identification, time and line and vehicle identification	Samples available only to EUBra-BIGSEA partners.
Census area data	Census information, such as population, income and literacy rate.	Made available by Instituto Brasileiro de Geografia e Estatística (IBGE, http://www.ibge.gov.br), which is the official curator of census data in Brazil.

Table 1: Data sources description availability for People Paths application

3.1.2 Installation and requirements

Requirements and installation instructions are described in the project repository.

3.1.3 Model construction

In order to perform trips origin/destination locations analysis, user trips must be identified using GPS and ticketing data. This is achieved using the following approach: Ticketing data is matched with bus GPS data, specifying where the user took the bus, and the matched data is grouped by user. Then, we select the users which took more than one bus in the given day. Then, we consider the user ticketing records as a time-ordered circular list of origin/destination locations, where the second location will be the destination of the first trip with origin at the first location, and so on; and the last location will be the source of the last trip back to the first location.

In the last step, origin/destination locations are matched with census area data, including population, income, and literacy rate.

Data sources and the process of model construction are shown in Figure 1. A lower level implementation, using Lemonade workflows, presented in Figure 2, shows each task of data transformation. All data are read from files stored in a local file system (*prototype version)* or in a distributed file system (HDFS) in the final version. Each intermediary stage produces new data that are stored again in the file system. In the case of Spark implementation, just some tasks produce intermediary data because some operations are *lazy* and are executed only when data are needed (in Figure 2, only tasks Create Visualization and Data Writer produce intermediary data).

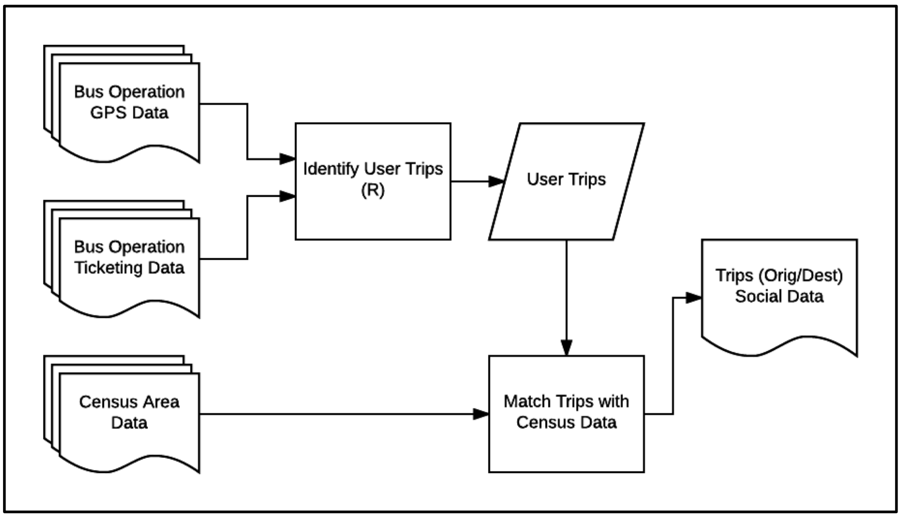

Bus Operation GPS Data
Bus Operation Ticketing Data
Census Area Data
Identify User Trips (R)
User Trips
Match Trips with Census Data
Trips (Orig/Dest) Social Data

Figure 1: Data sources and stages in People Paths application

3.1.4 Current state of development

We have implemented three versions of People Paths using different technologies and programming abstractions. All three versions are used in different phases of EUBra-BIGSEA and have the same workflow shown in Figure 2.

The first version was developed using R language and used a subset of data available in order to generate the model and evaluate it. This version is openly available at https://github.com/analytics-ufcg/people-paths.

The second version was implemented using PySpark, a Python language abstraction to the Apache Spark distributed processing platform. Apache Spark is one of the recommended tools by WP4 for Big Data processing. The source code, compatible with Apache Spark version 2.0.1, is available at: *https://github.com/eubr-bigsea/Lemonade_apps/blob/master/people_paths/spark_code.py.*

The third version is under development as part of the efforts to implement the Lemonade platform. Lemonade is a visual workflow construction tool, used as a programming abstraction proposed in WP5.

New data sources may be analysed together with existing ones such as bus stops, types of bus stops, bus terminals, POIs and types of buses. Bus terminals may be used as more likely destinations depending on the bus line type. For example, Curitiba City has a bus line type called “Expresso Ligeirão” (*“Fastest Express”* in Portuguese) that has reduced number of stops[1].

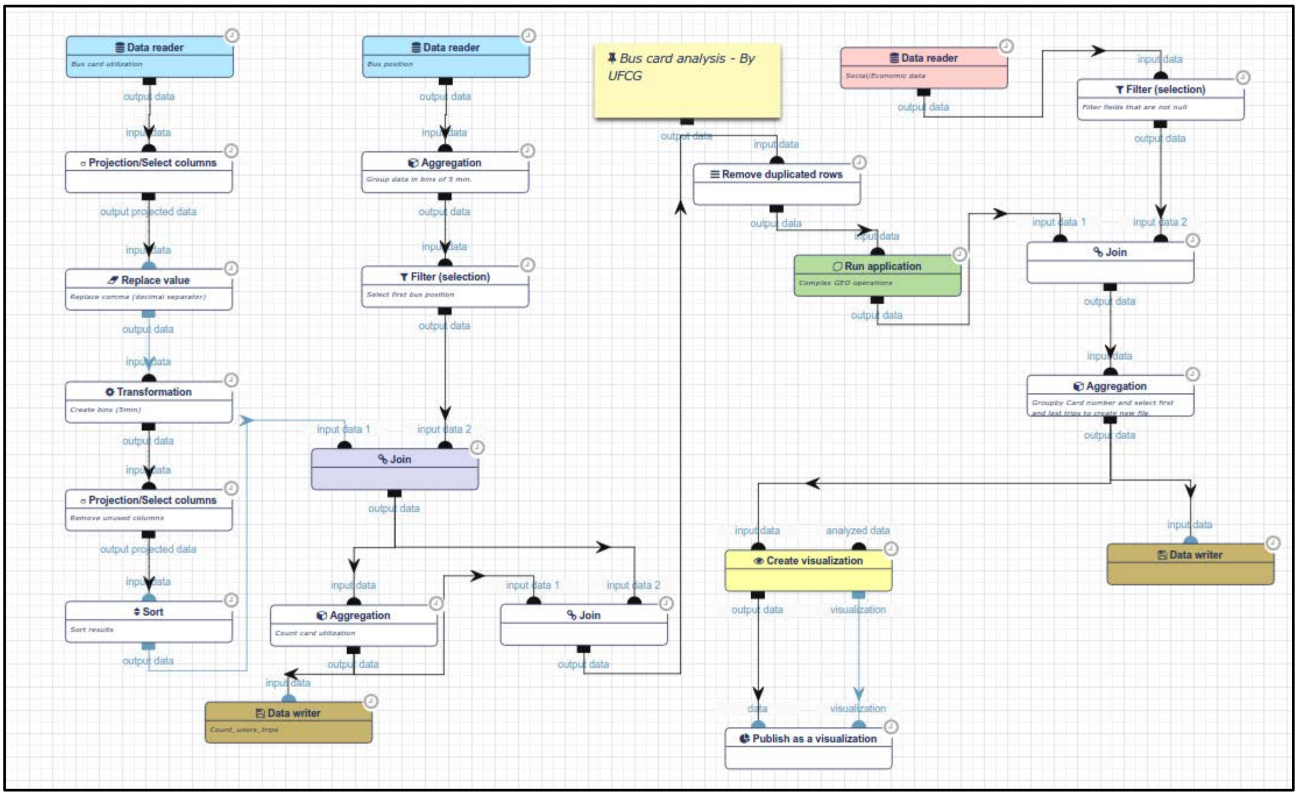


Figure 2: People paths application (WP7 application) represented as a Lemonade workflow

3.1.5 Usage in the EUBra-BIGSEA project

Context	Description of usage	Status and future work
WP4	Used to validate infrastructure installation and milestone MS8: First release of the big and fast data cloud platform.	Deployed and running in test environment. Needs to evaluate performance.
WP5	Implemented using abstractions proposed in the work package (Lemonade platform). Adaptation required introduction of new geo operations, such as *within* and *read shapefile.*	Application designed. Source code generated and tested in development environment.
WP6	Join operation between ticketing data and bus GPS data, performed in the application, is one of the scenarios being analysed by privacy and security group.	Scenario has been discussed with people from different WPs. WP6 working group will propose abstractions to Lemonade platform and primitives in WP4 in order to guarantee privacy and security constraints.
WP7	Result will be used in recommendation	Next step: Integrate the complete

[1] Bus line types are available at https://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte/24

	application (D7.5), Routes for People.	application.

Table 2: Usage of People Paths application in other EUBra-BIGSEA contexts

3.1.6 Sample result

Sample record, encoded in JSON format, is presented below. For each bus ticketing card, two new sets of information are incorporated and are related to the origin (*o prefix)* and destination (*d prefix).* Notice that attributes such as *o.pop, o.income and o.num.literate* are data from Census data. Actually, there are several other socio-economic information (described in https://goo.gl/qrJdDB) that may be used from Census data, such as age distribution, marital status, gender distribution of the inhabitants .

```
[{
  "card.num": "000001",
  "line.code": "260",
  "o.sector.code": "410690205040042",
  "o.neigh.code": "410690205033",
  "o.neigh.name": "SAO LOURENCO",
  "o.loc": "-25.388173,-49.260771",
  "o.timestamp": "2016-06-25 19:43:55",
  "o.pop": "833",
  "o.income": "3268.35",
  "o.num.literate": "781",
  "d.sector.code": "410690205040042",
  "d.neigh.code": "410690205033",
  "d.neigh.name": "SAO LOURENCO",
  "d.loc": "-25.388173,-49.260771",
  "d.timestamp": "2016-06-25 19:43:59",
  "d.pop": "833",
  "d.income": "3268.35",
  "d.num.literate": "781"
}]
```

Figure 3 presents a graph visualisation using generated data from one weekday analysis. Each edge connects an origin to the respective destination for a trip. Nodes are the different *census tracts* (one of the ways to segment cities, according to IBGE[2]). Node size is proportional to its in-degree value.

[2] http://www.ibge.gov.br/

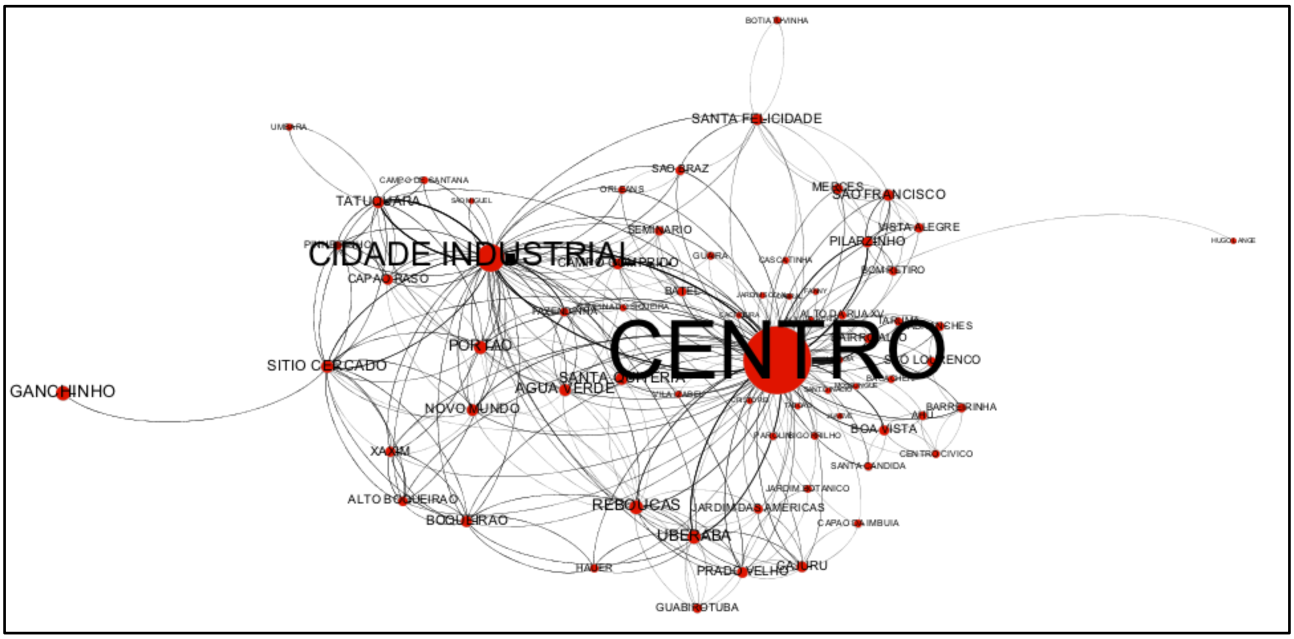


Figure 3: Graph representation of origin and destination in routes identified by model

3.2 City Administration Dashboard

City Administration Dashboard is an application that uses descriptive statistics and visualization techniques based on historical data of bus trips of a Public Transport System in order to assist and facilitate planning and monitoring the system.

Available at https://github.com/analytics-ufcg/City-Administration-Dashboard.

3.2.1 Description

This dashboard application compares scheduled and actual trips. Obtaining such data requires two steps of data processing. The first one is to identify all performed trips on execution data, e.g. GPS data. The second step is to match the actual trips to the scheduled trips. This process is described in following sections.

3.2.2 Installation and requirements

Requirements and installation instructions are described in project repository.

3.2.3 Model construction

Finding trips on execution data

In order to split execution data into trips, we use an algorithm that measures the similarity of a sequence of GPS coordinates to the shapes of all routes on the system.

Pairing performed and scheduled trips

This task uses essentially the starting time of both trips. The scheduled trip is paired to the performed trip with the closest start time, as long as the difference between them do not exceed the scheduled trip headway. The headway of a scheduled trip is the time difference, in minutes, between its start time and the start time of the next scheduled trip.

The data processing flow is shown in Figure 4. A small web application plus an API were developed in order to test and evaluate the application.

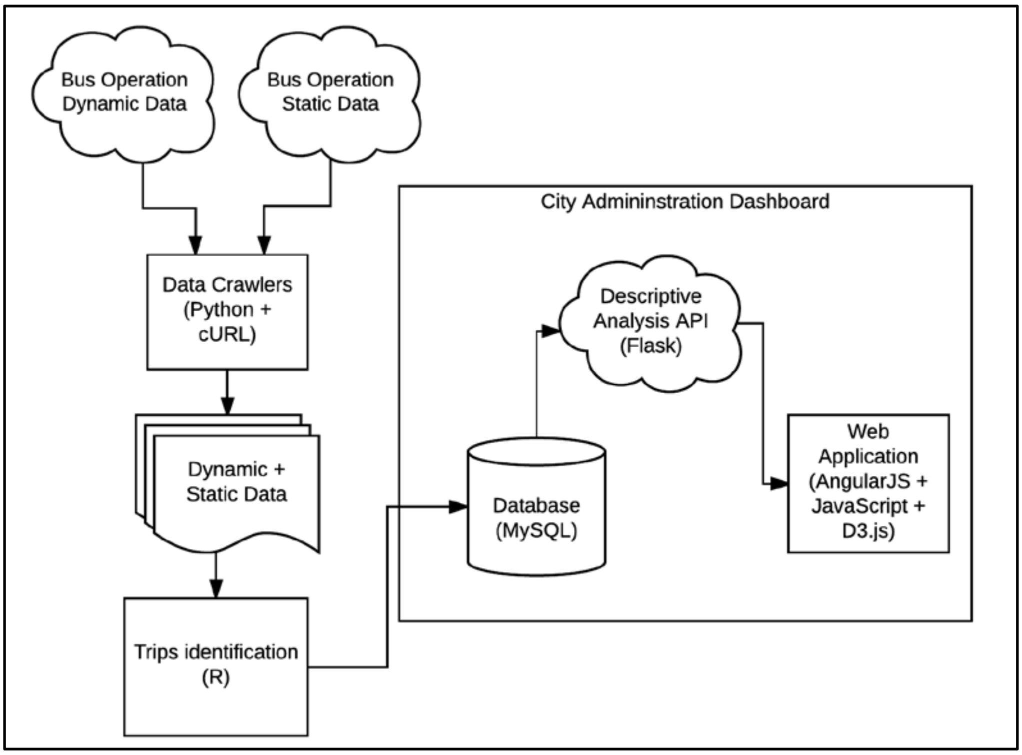

Bus Operation Dynamic Data
Bus Operation Static Data
Data Crawlers (Python + cURL)
Dynamic + Static Data
Trips identification (R)
City Admininstration Dashboard
Descriptive Analysis API (Flask)
Database (MySQL)
Web Application (AngularJS + JavaScript + D3.js)

Figure 4: Data processing flow and deployment of City Administration Dashboard application

3.2.4 Current state of development

Experimental version implemented in Python language. It needs to be ported to WP4/WP5 technologies.

3.2.5 Usage in the EUBra-BIGSEA project

Context	Description of usage	Status and future work
WP4	Used to validate infrastructure installation and proposed technologies.	Integration with Ophidia Analytical tool (under development).
WP5	Implementation in COMPSs.	To be initiated.
WP6	Integration with privacy policies with regard trips identification.	To be integrated.
WP7	Integration of results with dashboard application under development by UPV.	Next step: Integrate the complete application.

Table 3: Usage of City Administration Dashboard application in other EUBra-BIGSEA contexts

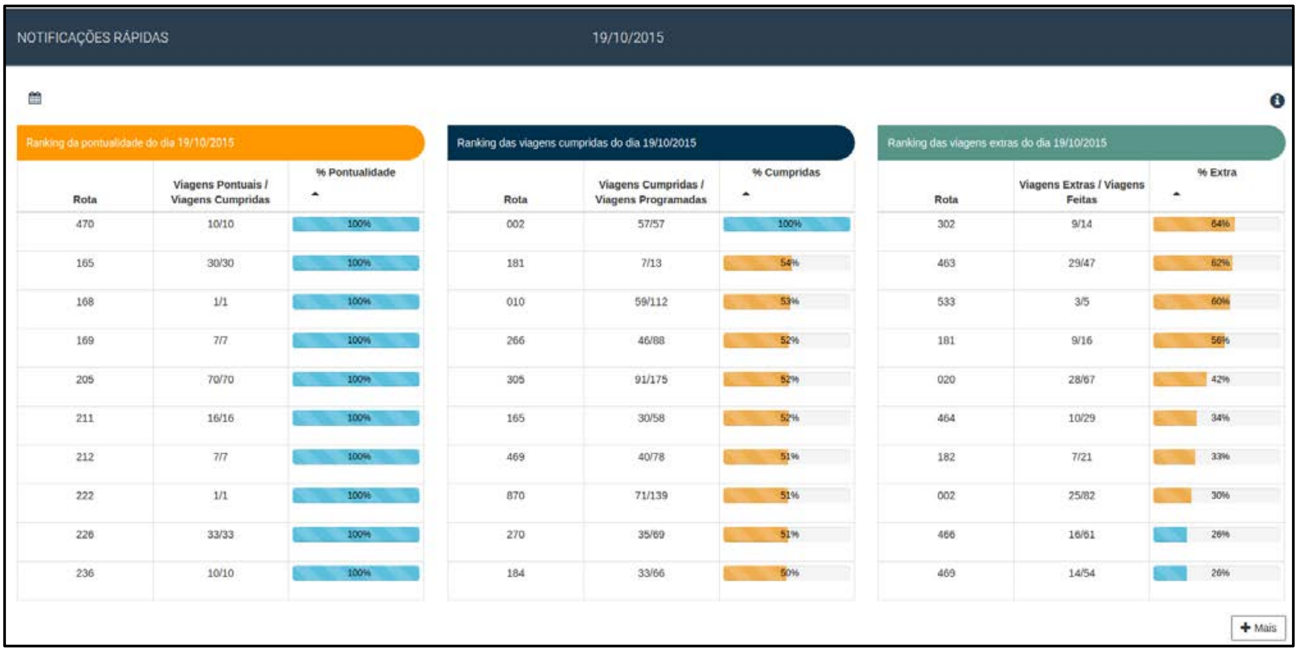
3.2.6 Sample result

A prototype was built and it is available in application repository. Below we describe some features implemented.

Rankings

Figure 5 shows the dashboard visualizations based on a ranking of the bus routes (sometimes referred as bus lines). The intention is to provide a fast and simple way to identify whether a bus route is according to the schedule.

The left panel shows the ranking of bus route punctuality. Route punctuality is basically the percentage of performed trips that were performed without delays. The middle panel shows the schedule fulfillment, i.e., how many scheduled trips were actually performed. Finally, the right panel presents the number of *extra trips*, i.e., how many performed trips did not have an associated scheduled trip.



NOTIFICAÇÕES RÁPIDAS

19/10/2015

Ranking da pontualidade do dia 19/10/2015

Rota	Viagens Pontuais / Viagens Cumpridas	% Pontualidade
470	10/10	100%
165	30/30	100%
168	1/1	100%
169	7/7	100%
205	70/70	100%
211	16/16	100%
212	7/7	100%
222	1/1	100%
226	33/33	100%
236	10/10	100%

Ranking das viagens cumpridas do dia 19/10/2015

Rota	Viagens Cumpridas / Viagens Programadas	% Cumpridas
002	57/57	100%
181	7/13	54%
010	59/112	53%
266	46/88	52%
305	91/175	52%
165	30/58	52%
469	40/78	51%
870	71/139	51%
270	35/69	51%
184	33/66	50%

Ranking das viagens extras do dia 19/10/2015

Rota	Viagens Extras / Viagens Feitas	% Extra
302	9/14	64%
463	29/47	62%
533	3/5	60%
181	9/16	56%
020	28/67	42%
464	10/29	34%
182	7/21	33%
002	25/82	30%
466	16/61	26%
469	14/54	26%

Figure 5: Bus routes rankings in dashboard application

Routes daily operation

This functionality allows users to investigate the operation of one individual bus route. A feature, called "Escala" (*scale* in Portuguese), displays all performed and scheduled bus trips using an arch-based representation.

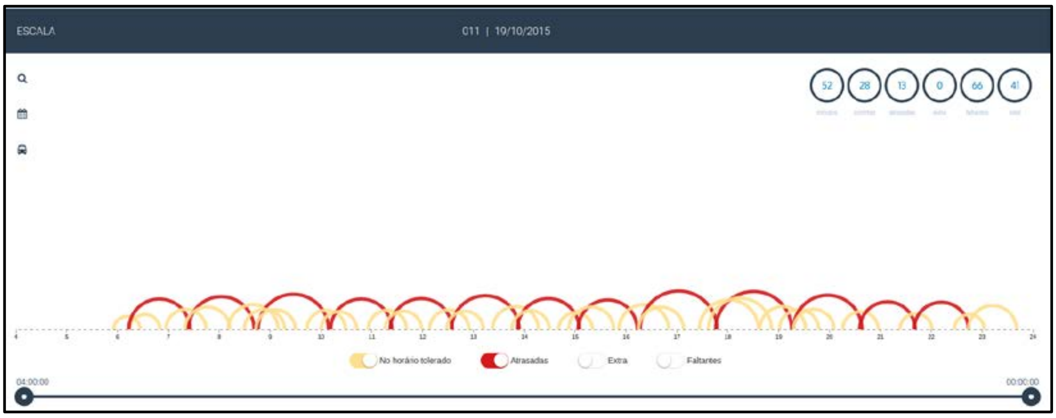


Figure 6: Bus routes rankings in dashboard application

In Figure 6, the *x* axis represents the hour of the day. Each arch color represents a different status for a bus trip: trip has been performed on time, trip was late, trip was not scheduled (extra trip) and trip was not performed (missing).

3.3 Topic detection in online social network data

This descriptive model tries to identify topics being discussed in Twitter OSN. Identified topics may be used to infer the cause of events in the context of smart cities. For example, a traffic jam may be occurring because people are protesting in a region or because there is a flood caused by rain.

Available at https://github.com/eubr-bigsea/topic-detection (under development).

3.3.1 Description

Topic identification methods are among the most explored tools to extract information from large amounts of data. They were conceived to find semantically meaningful topics from a document corpus, and they assume that there are hidden variables (topics) that explain the similarities between observable variables (documents). These techniques are usually based on one of the following approaches:

- Probabilistic: These methods assume that the data was generated by a generative model that includes the hidden variables. This generative process defines a joint probability distribution over both the observed and hidden random variables that allow us to infer the existing topics;
- Non-probabilistics: These methods are usually based on matrix factorization techniques, where the matrix terms X documents, which represents the database, is projected into a k-dimensional space where each dimension is a topic.

The main representatives of the aforementioned approaches are Latent Dirichlet Allocation (LDA) [R5] (probabilistic) and Non-negative Matrix Factorization (NMF) [R6] (non-probabilistic). In summary, these algorithms rely on the words co-occurrence in the document level to infer the topics.

3.3.2 Requirements

Requirements and installation instructions are described in project repository.

3.3.3 Model construction

Latent Dirichlet Allocation (LDA) [R5] is a method conceived to describe document collections using a set of topics. The most intuitive way to understand its rationale is by looking at its generative process, i.e., an imaginary process assumed by the model when the documents are created. A topic is formally defined as a probability distribution over a fixed vocabulary. For example, the topic sports can be described by a probability distribution concentrated on terms such as game, football, tennis, ball, and match.

Let us assume there exists a set of topics predefined before the documents are generated. In this case, for each document in a collection, its terms are generated in two phases:

1. Choose Θd, a random multinomial distribution over the predefined topics
2. For each term Wd,n in the document
   a. Choose Zd,n, a topic from the topic distribution Θd
   b. Choose a term from the topics chosen in 2(a)

This statistical model reflects the fact that documents contain multiple topics. For example, a document may match sports and politics. However, each document has topics in different proportions (defined in phase 1), and each term of the document is chosen from one of the topics (phase 2b). The selected topic, in turn, is chosen from a topic distribution for the document being generated (phase 2a).

This generative process may be represented by the graphical model below, where nodes are random variables, edges indicate dependence, plates indicate replicated variables and shaded nodes are the observed nodes.

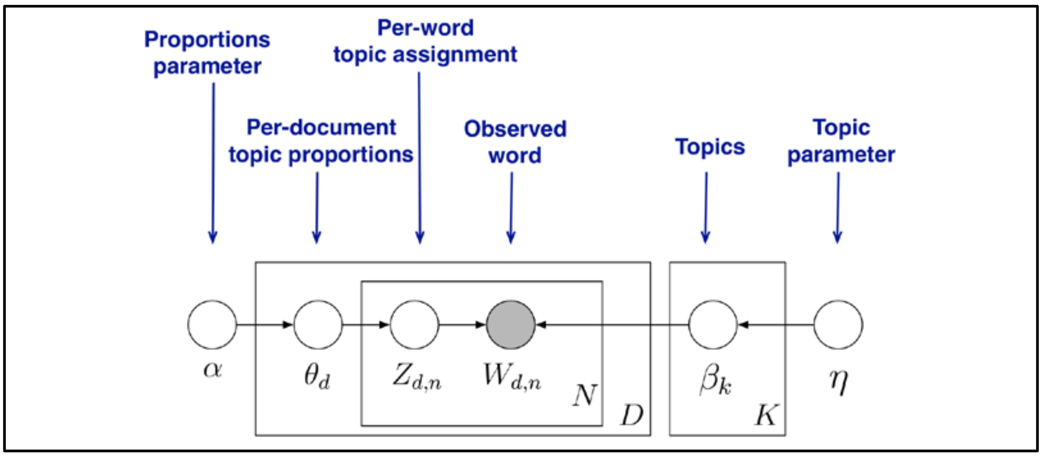


Figure 7: LDA generative process

As we have the documents and their words, the real problem addressed by the LDA algorithm is how to learn the other variables, including per-document topic proportions, per-word topic assignments and topics distribution. This is achieved by an inference algorithm, such as the Expectation Maximization (EM) [R7], Maximum Posteriori Estimation (MAP) [R8], Gibbs Sampling [R9] or Online Learning [R10]. The most popular algorithm used in the literature is, by far, the Gibbs sampling, but Apache Sparks supports only EM and Online Learning in the DataFrame API.

Input data

Experiments use a dataset with 100000 *tweets* (posts in Twitter OSN). Each *tweet* is identified by a number and contains besides this number, the publication date and its text. Text may contain 1 to 140 characters.In the dataset, the average length was 56 and after removing punctuation and *stopwords*, the final average length was 40. Data were obtained during 7 days, starting in December 22th, 2016. All *tweets* contain user localization, with best precision provided by GPS information and in the worst case, the precision is the just the city. We filtered only data associated with users from the city of Curitiba.

In this experiment we were interested in one of the challenges in topic identification: how to uncover the topics in short text documents. Extracting topics from short text is difficult because of the dependence of the methods in word co-occurrence, which in short text are rare and make conventional algorithms suffer from severe data sparsity [R12]. To increase the vocabulary and number of features in each document, we decide to convert text into *bigrams* before vectorizing it.

Vector Representation of Words

Most traditional natural language processing techniques consider words as atomic units of processing [R11]. In the experiment, we are using bigrams as unit of processing. It is part of the this work test other configurations of units of processing and vector representation, such as Continuous Bag of Words (CBOW) [R11] and Skip-Gram (SG) [R11]. Both models generate the representations using neural networks.

Algorithm configuration

The LDA algorithm has four main parameters:

1. k: number of topics. Here, we tested the algorithm with 50;
2. alpha: the hyper-parameter alpha for the dirichlet distribution. We use the default value suggested by the authors: 50 over the number of topics;
3. beta: the hyper-parameter beta for the dirichlet distribution. We use the default value suggested by the authors: 0.1;
4. niters: number of iterations of the algorithm. We use the default value: 2000.

Data processing workflow

The final version of data processing workflow is shown in Figure 8. As mentioned, input data are formed by *tweets.* We are also using another dataset for Portuguese *stopwords.* The flow follows after reading data, removing punctuation and accents, breaking text into bag of words, removing *stopwords*, generating bigrams, vectoring them and finally running the LDA algorithm and generating the final report with topics identified.

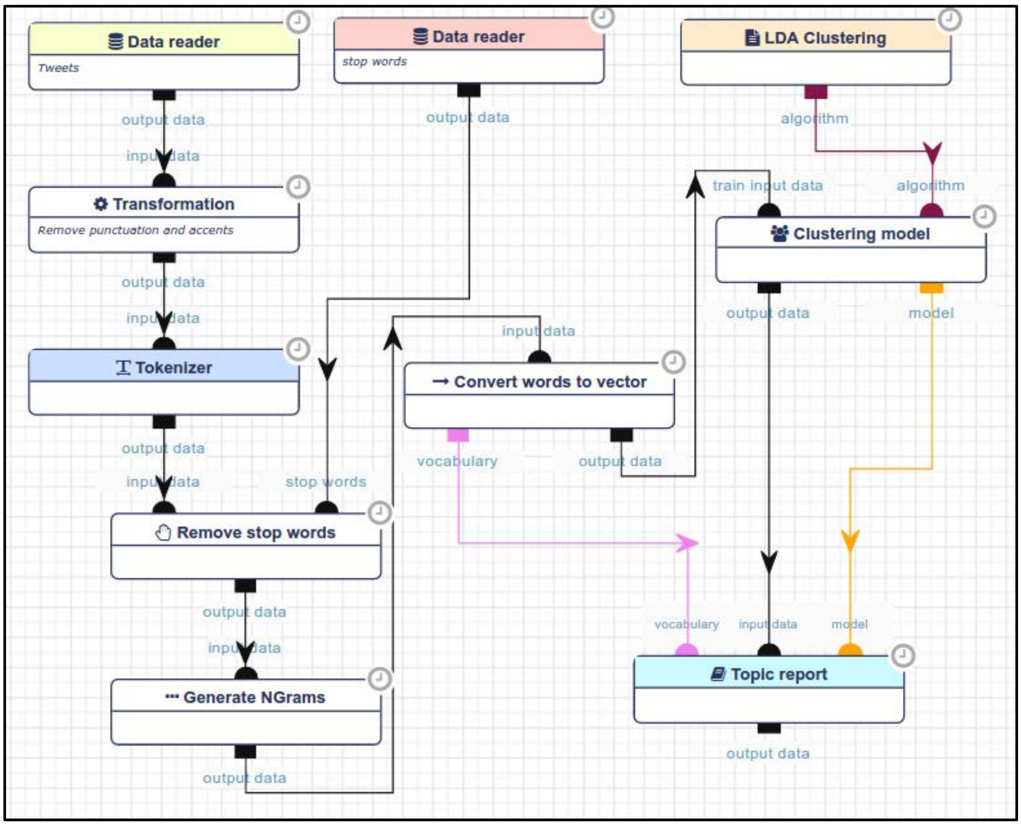


Figure 8: LDA generative process

3.3.4 Current state of development

While the approach of considering words as atomic units has produced impressive results in many domains, it can be improved with the exploitation of word meanings and similarities, especially for rare terms. Symbolic representation of words do not take into account these issues, except in cases where morphological similarity captures semantics, but this is far from being a rule, and it is frequently a source of ambiguity (e.g. synonyms and homonyms).

LDA performs better when there are a large number of features (words) in a document. In the case of *tweets* and other short documents, there are alternatives that may be evaluated. Latent feature LDA - LF-LDA [R14] and Biterm Topic Model - BTM [R13], were proposed to deal with short text. There are open-source implementations of both and they could be ported to project technologies.

Finally, results should be evaluated in some way. Evaluating topic modeling methods is not a straightforward process. Two metrics are well accepted in the literature: (i) NPMI-Score [R15] and (ii) topic coherence [R15] and will be integrated soon into the toolbox.

3.3.5 Usage in the EUBra-BIGSEA project

Context	Description of usage	Status and future work
WP4	Integration with Spark and HDFS data (large volume of OSN	Under development. Alpha

	data)	version implemented in Spark using Lemonade Platform. Implement *stream* processing.
WP5	Used as a sample application in order to identify needed operations to process textual data in Lemonade Platform. New operations identified include *remove stop-words, remove accents, tokenize strings and convert text to vector.*	New operations implemented in Lemonade Platform. Application designed in Lemonade.
WP6	According to Twitter API usage policies, data (*tweets)* shall not be distributed and, if demanded, they should be removed from datasets. For example, once a *tweet* is excluded, it should be excluded from previously obtained data.	Evaluate the requirements to implement this integration with data source provider.
WP7	Integration with city administration dashboard	To be done.

Table 4: Usage of topic identification model in other EUBra-BIGSEA contexts

3.4 Traffic Congestion Detection using Taxi Position

Traffic congestion is a frequent situation in urban centers nowadays. It is a consequence of the urban infrastructure not being able to keep up with the growth of the number of vehicles. Thus, it causes many drawbacks, such as stress, delays, and excessive fuel consumption. This application aims to identify traffic congestions using the geographic position of taxis provided by GPS. We assume that a vehicle speed may be estimated by its position in different times. Thus, we applied a density clustering algorithm, which takes into account both spatial and non-spatial aspects [R3], to identify traffic congestions from taxi positions.

Available at https://github.com/eubr-bigsea/py-st-dbscan.

3.4.1 Description

Input data

The dataset was provided by a Brazilian taxi company called *"Vá de Táxi"*. Each record represents the position of one taxi vehicle in a specific date and time. There are 511,736 positions, generated by 653 unique taxi drivers on May 17th, 2016. Dataset attributes include taxi position (latitude and longitude), date and time, and the anonymous identifier number of the taxi driver. Taxi positions are collected every minute, through the drivers' smartphones, even if the taxi is not occupied by a passenger in a ride.

A database sample was provided by "Vá de Táxi" to the PUC-MG University and UFMG was allowed to access it in EUBra-BIGSEA project under the condition of not distributing it.

3.4.2 Requirements

Requirements and installation instructions are described in the project repository.

3.4.3 Model construction

The model applied is the ST-DBSCAN [R4] in order to identify traffic congestions based on the vehicle position at different timestamps. As taxi positions are collected every minute, it is possible to infer their speed considering their position variation. We are assuming that if a vehicle is moving slowly, then it is on a congested road.

Based on this assumption, the ST-DBSCAN was adapted in order to identify traffic congestion. Such adaptation was done in the selection of a new element for a cluster. Thus, one element belongs to a cluster if it has the minimum distance and it is in the minimum time interval of another element already in this cluster. The cluster average was not considered otherwise large congestions would not be identified. After identifying all clusters, the distance between every pair of elements in the same cluster is computed. We call this distance amplitude. Only clusters presenting the maximum amplitude distance greater than a threshold were kept. This process is important for removing clusters that represent places where vehicles were stopped in semaphore or waiting for passengers.

The approach for identifying traffic congestions has four parameters. The first three parameters are ST-DBSCAN parameters and the last one is related to post-processing:

1. **Eps1** defines the function that limits the neighborhood area. The value of Eps1 must be neither too short, otherwise it would include stopped or parked vehicles, nor too large, otherwise it would not consider vehicles with low speed in congested regions. The values chosen for Eps1 were 100, 166 and 330 meters, which are equivalent to 6, 10 and 20 km/h respectively.
2. **Eps2** also restricts the neighborhood, but it considers temporal aspects. Eps2 was set as 60 seconds, so that it is aligned with the time interval which taxi positions were collected.
3. **minPts** defines the minimum number of neighbors that a point must have to be considered a member of a group. Otherwise this point is considered noise. Its value was fixed in 2 because clusters have linear shapes and a value too large for this parameter could classify clusters as noise erroneously.
4. **minAmp** is the threshold that defines the maximum amplitude for the cluster not be classified as a stopping point (e.g., taxi kept in the same position waiting for passengers). The parameter *minAmp* was set as 500 meters.

3.4.4 Current state of development

Model generation is implemented as a research prototype. It is written in Python language and demands more tests, performance tuning and evaluation using different input data.

3.4.5 Usage in the EUBra-BIGSEA project

Context	Description of usage	Status and future work
WP4	Integration with Spark and HDFS data (other data sources, such as bus GPS data and Waze data). Include this library in the WP4 list of libraries and technologies incorporated.	This application introduces an example of utilisation of the *clustering* technique data mining/machine learning. It may present challenges in its migration process to WP4 technologies, specially related to scalability.
WP5	Implement ST-DBScan algorithm as an operation in Lemonade Platform.	New operations implemented in Lemonade Platform.

WP6	Application uses non-public data and results should contain just aggregated information.	Integrate with AAA services (to be initiated).
WP7	Integration with city administration dashboard in order to show areas facing traffic jam.	To be done.

Table 5: Usage of application in other EUBra-BIGSEA contexts

3.4.6 Sample result

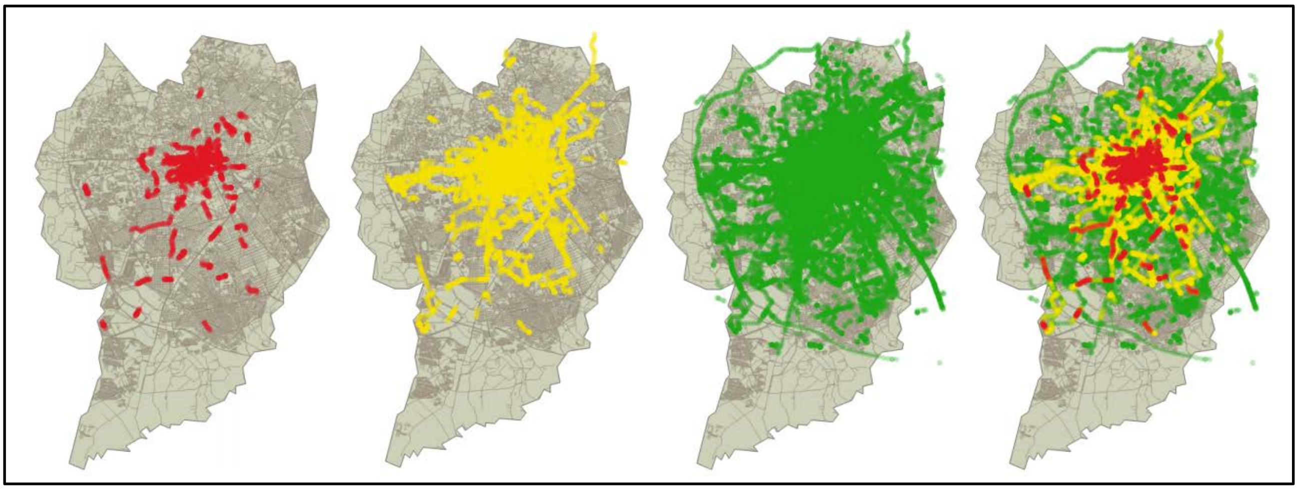


Figure 9: Map plotted result for different values for parameter Eps1: (a) 100m, (b) 166m, (c) 330m and (d) all together, with fixed parameters Eps2=60s, minPts=2.

Figure 9(a), 9(b) and 9(c) shows clusters with radius equal to 100, 166 and 330, respectively. Figure 9(d) shows all the clusters.

In order to produce Figure 1(a), the parameters were chosen to capture vehicles traveling with at most 6km/h. Thus, in this figure, there are dense and small clusters. In total, it was identified 123 clusters with an average distance equal to 1,045 meters. As 6km/h is a speed too slow for a vehicle, either the taxi was traveling too slow or the taxi driver got out of his vehicle with his application turned on.

In Figure 1(b), it is shown larger clusters. The number of clusters is 484 and their average distance has doubled to 2,061 meters. This case represents taxis with speed at most 10km/h, slow speed for a vehicle and fast speed for a person by foot.

On the other hand, Figure 1(c) shows clusters over the entire city of Curitiba. The Eps1 fixed to 330 meters produces clusters of vehicles with speed of 20km/h, which is a normal speed for many roads in the city and it does not necessarily indicate slow traffic. In this way, were identified 9.522 clusters and their average distance was 1.178 meters.

Finally, in Figure 1(d), clusters generated by all values of Eps1 were shown in order to show a global vision of traffic during a day in Curitiba. As expected, for all values of Eps1 the downtown of Curitiba has a large concentration of clusters, due to the large number of people and commercial buildings. On the contrary, the other areas show lighter traffic intensity as the vehicles present greater average speed.

4 CONCLUSIONS

In this report we present the principles and strategies for building descriptive models in the scope of the EUBra-BIGSEA project. We discuss how descriptive models are built and their usage in the scope of smart cities applications. We also present four models constructed and evaluated in the scope of the project as well as their applications to real scenarios. As next steps, we will continue implementing novel relevant models on the EUBra-BIGSEA eco-system exploiting the full stack developed in the project (from WP3 to WP6) and also assessing to what extent the Lemonade environment is effective in constructing the models and associated applications.

5 REFERENCES

[R1] Heldal, Rogardt; Pelliccione, Patrizio; Eliasson, Ulf; Lantz, Jonn, Derehag, Jesper; Whittle, Jon . 2016. Descriptive vs prescriptive models in industry. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS '16). ACM, New York, NY, USA, 216-226. DOI: https://doi.org/10.1145/2976767.2976808

[R2] Taylor, K. L. (2014). Oracle Data Mining Concepts Retrieved Oct. 1, 2015, from https://docs.oracle.com/database/121/DMCON/E17692-15.pdf

[R3] Birant, D. and Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1):208 – 221. Intelligent Data Mining.

[R4] [Birant and Kut, 2007] Birant, D. and Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1):208 – 221. Intelligent Data Mining.

[R5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. J. Mach. Learn. Res., 3, March 2003.[R5]

[R6] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–791, 1999.

[R7] Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, May 2014. ISBN: 9780521766333.

[R8] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[R9] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101(suppl 1), 5228-5235.

[R10] Hoffman, M. D., Blei, D. M. & Bach, F. R. (2010). Online Learning for Latent Dirichlet Allocation.. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel & A. Culotta (eds.), NIPS (p./pp. 856-864), : Curran Associates, Inc..

[R11] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[R12] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), 12, 1532-1543.

[R13] Yan, Xiaohui, et al. "A biterm topic model for short texts." Proceedings of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2013.

[R14] Nguyen, D. Q., Billingsley, R., Du, L., & Johnson, M. (2015). Improving Topic Models with Latent Feature Word Representations. Transactions of the Association for Computational Linguistics, 3, 299-313.

[R15] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Advances in neural information processing systems (pp. 288-296).

GLOSSARY

Acronym	Explanation
AAA	Authentication,Authorization and Accounting
API	Application program interface
BTM	Biterm Topic Model
CBOW	Continuous Bag Of Words
EM	Expectation Maximization (algorithm)
GES^3	Georeferenced Environmental, Stationary, Streaming and Social
GPS	Global Positioning System
HDFS	Hadoop Distributed File System
IBGE	Instituto Brasileiro de Geografia e Estatística (Brazilian organization responsible for Census data)
JSON	JavaScript Object Notation
LDA	Latent Dirichlet allocation (a generative statistical model)
LF-LDA	Latent Feature - Latent Dirichlet allocation
NMF	Non-negative Matrix Factorization
NPMI	Normalized Pointwise Mutual Information
OSN	Online social networks
POI	Point of Interest
PUC-MG	Pontifícia Universidade Católica de Minas Gerais (Brazil)
ST-DBSCAN	Spatial-Temporal - Density-Based Spatial Clustering of Applications with Noise