IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 21, 2021, accepted August 9, 2021, date of publication August 25, 2021, date of current version September 22, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3107833*

# Integration of Automated Vehicle Location, Fare Control, and Schedule Data for Improved Public Transport Trip Definition

**JUAN BENAVENTE, BORJA ALONSO, ANDRÉS RODRÍGUEZ, AND JOSÉ LUIS MOURA**
SUM+LAB, Department of Transportation and Technology of Projects and Processes, University of Cantabria, 39005 Santander, Spain

Corresponding author: Juan Benavente (juan.benavente@unican.es)

This work was supported in part by the Ministerio de Ciencia Innovación y Universidades through the European Regional Development Fund under Project TRA2015-69903-R, in part by the EU Horizon 2020 Project An open, sustainable, ubiquitous data and service ecosystem for efficient, effective, safe, resilient mobility in metropolitan areas (SETA) under Grant 688082, and in part by Santander City Council [Joint development of an advanced mobility model for the city of Santander based on big data techniques and tools (MOBIDATA-SDR)].

**ABSTRACT** This paper proposes a flexible methodology to improve the definition of each distinct trip carried out in a transport system, integrating the information provided by stop-level events from its automated vehicle location and fare collection systems, and scheduling subsystem information at the initial stop of planned trips. The data are structured; and then corrected and completed utilizing several criteria, including a probabilistic approach based on the distributions of travel and dwell times, aiming to minimize the distortions that appear due to the nature of the available sources. The case study data encompass one year of records from the automated vehicle location, fare collection, and scheduling subsystems in Santander City, Spain. The results are discussed with captures from an interactive web visualization tool that has been developed for this work.

## HIGHLIGHTS

- Trips that have taken place are recognized, providing for each call arrival and departure times, as well as identifying the raw data sources that were utilized to make the determination.
- Each cluster of ticketing events is assigned to the corresponding visit of a vehicle.
- Distinction between trips that are part of the planned timetable, and those that respond to operational decisions.
- Detection and treatment of instances where the id of a vehicle changes during a trip.
- Robustness against missing and erroneous data.
- Specific treatment for particularly problematic termini.

**INDEX TERMS** Data integration, public transportation, smart cards.

## I. INTRODUCTION

Multiple opportunities for research and development stem from the analysis of the current implementations and future possibilities of intelligent public transportation systems (IPTSs), being the integration of data from different subsystems to create better models one of them [1], [2]. However, a series of common problems arise: some related to each separate source, others to the fusion process. These include missing, redundant or erroneous entries; fragmentation of the sequences of stops that are part of a single trip; clocks of different devices not synchronized; or inconsistent ids for the same elements along different tables.

One of the most relevant features for the analysis of a public transit network is the characterization of the trips as they are carried out during day-to-day operations. Each of these can usually be defined by one of a previously established set of routes, and the arrival and departure times from each of the pre-defined stop locations. The datasets which are useful to build this transportation offer model may, in most cases, be obtained from the IPTSs.

The method described in this work originated from the necessity to create a representation of the public transportation supply using boardings-only automated fare

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.

control (AFC) and automated vehicle location (AVL) stop-level data, and the planned (and sometimes recorded) starting times of the trips from an IPTSs where the aforementioned problems made the available dataset initially too fragmented, incomplete, and inaccurate for its intended uses (user behavior modeling, inference of bus load profiles, and obtaining transit origin-destination matrices).

The case of Santander (Spain, 173 957 inhabitants) is studied; utilizing data that encompass the AVL and AFC events of one year, and the schedule of where and when each planned trip begins.

This methodology can be synthesized in these steps: preprocess AVL and AFC input data, obtaining tables where each visit of a vehicle is represented by a single entry; identify the sequences of bus stops that completely carried out, perfectly recorded trips of each route must follow; classify AVL information in fragments of these sequences; choose travel and dwell times models for each route; build tables that describe when each bus trip visits each stop, combining AVL and AFC entries; detect and treat instances where the id of a vehicle changes mid-trip; link trips to planned starts, making use of the extra information to improve their characterization; attribute ticketing events to the proper visit of a bus during a trip; and finally filter out trips not supported by enough IPTS evidence.

The procedure described in this paper aims to be suitable in situations with different information availability, completeness and reliability. Particularly, scheduled trip beginnings may be known or not, optionally including which vehicle had been initially assigned to the task. Available AVL, AFC, and planning subsystem data are combined to create a better and more useful characterization of the trips that have taken place in a transportation system: arrival and departure of the vehicles and their corresponding cluster of ticketing events for each stop of each trip; and distinction between those trips that materialize the planned timetable and those that occur due to operational decisions.

An interactive web-based visualization tool has been developed to show the improved definition of the trips carried out in the transport system, based on the Bokeh Python library. It has been utilized to illustrate the methodology (figs. 5, 6, 8 and 9) proposed in this work, and its application to the case study (figs. 12 to 14). It signifies the data source that has been used to deduce each visit of a vehicle, and the search ranges computed to complete each trip; and allows to dynamically choose which vehicles to show, and to vary the temporal span being represented.

## II. LITERATURE REVIEW

Stop-level records, which can be stored in a IPTS at a fairly low incremental cost, have allowed to better estimate previously utilized performance indicators and usage metrics (e.g. travel times) [3], and also to assess previously nigh impossible to quantify attributes due to data scarcity, such as those related to service reliability [4]. However, they may require a significant effort to attain meaningful conclusions [5]. Also, adequate visualization tools are needed to be able to comprehend the vast amount of output that can be generated [6].

Setting aside those cases where fixed timetables are not available (e.g. bus routes in Jinan, China, with high uncertainty in travel times, multiple agencies, and a departure schedule that changes according to on-site observed demand, where a study employed artificial neural networks for improved real-time bus arrival estimation based on AFC and historical vehicle location information [7]), the data that describe transit services (i.e. routes, their schedules, and where the stops are) are published in advance. The most widespread tool to do so is the static component of the general transit feed specification (GTFS) [8]. However, even though an extension has been proposed [9], this format cannot yet represent some real-time changes, such as defining additional trips. Also, transportation agencies may not keep a compilation of these files through time, though in some cases they can be obtained from a third party (for example OVapi [10], Transitland [11], or OpenMobilityData [12]).

Besides other applications such as identifying headway irregularities [13], implementing more intelligent vehicle priority strategies [14], and fleet management and operations [15]; global navigation satellite system (GNSS) location information is used to initially estimate and finally identify the arrival of the vehicle to each point of interest. Typically, it offers a 5 m precision under open sky, though several factors can worsen its accuracy [16].

Since requiring to register a state where the bus is completely still next to a stop to assert that a visit is happening could require too frequent updates, with their associated network traffic; the actual arrival event is usually equated with the vehicle being detected inside a relatively small region that encloses the stop, while its velocity is lower than a threshold [17]. This event is stored in the AVL database, including besides its timestamp and bus stop identifier other possibly useful information obtained from on-board sensors: dwell time, route identification, door opening and close times, etc.

On the other hand, AFC systems have as main purpose to improve the revenue collection process, but they also provide valuable data, especially when enriched with the user tracking and characterization possibilities of smart card (SC) technology.

AFC information can be useful at the operational, tactical, and strategic levels of public transport management, with multiple applications [18] such as passenger behavior modeling [19]–[21], event-based multi agent simulation [22], vehicle load profiles [23], quality of service assessment [24], constructing transit origin-destination matrices [25], [26], or estimation of passengers' excess journey time [27]. Obstacles to fully benefit from this source of information are the reluctance of farebox manufacturers to ease communications with other on-board devices to prevent fraud, the disinclination of the operator to share business-sensitive details [28], and that a validation may not be required to exit

the system (''tap-in'' only configuration). For instance, in a study with data from Guangzhou (China) [29] researchers decided to develop a methodology to extract bus boarding and alighting information from access-only raw SC data that does not identify the stop where it happened, combining the identification of trip direction, boarding cluster, boarding stop, and alighting stop (utilizing a series of criteria that build upon the trip chaining theory [30]). Another example is the use of data from a ''tap-in, tap-out'' public transport network in Singapore [31] where researchers explore the reasons why AFC may provide incorrect information, and propose how to identify these erroneous entries and their source. Recently, scientists from Brisbane (Australia) and Hong Kong (China) have published a review in the field of transit OD estimation [32] where AFC data cleansing is identified as its first component, identifying sources and types of errors, and classifying boarding stop estimation problems based on which features are available in the SC data.

Each trip performed by a bus can be conceptualized as a path that starts at a first stop, continues as the bus calls at midway stops, and ends at a final one. From a spatiotemporal perspective, it can be regarded as a concatenation of sections [33], where each of them encompasses the time between arrivals at two (non-necessarily consecutive) stops; or as a series of calls at consecutive stops and traveling the links between them [34]. In the latter case, dwelling time at each stop depends on the number and characteristics (special needs and payment mode) of alighting and boarding passengers, how long door operations take, etc. [35]; while link travel times are affected by the available infrastructure, service management, traffic flows, driver behavior, weather, etc. [36].

Several probability distributions are proposed in the existing literature to characterize the variability of link travel times [34] such as shifted log-normal, log-normal, normal [36], gamma, Weibull, Burr Type XII [37], generalized extreme value [38], etc. Numerous real-life studies [34], [39]–[41] choose the former, which shows a probability density of zero when the value of the random variable falls below a threshold (which would be the free-flow link travel time) and can adequately fit asymmetric, positively-skewed data; and that for many links is the function that most likely describes how travel times are distributed. A 2017 study conducted on global positioning system (GPS) data from taxis during the morning peaks of 5 weekdays in Wuhan (China) [42] found that link travel times may be best represented by log-normal, gamma or normal distributions (on 50%, 30%, and 20% of the analyzed links, respectively) and opted, to avoid computationally intractable calculations, to assume that travel times along a path can be approximated by normal distributions.

Regarding dwell times, the majority of works suggest that, due to their non-negative nature and possible skewness, the log-normal distribution is likely to be the best alternative (e.g. a study of 18 months of data from a bus route in New Jersey, USA [43]; 6000 records from a one-day study in Changzhou, China; or an analysis of 1-month data from public buses in Jinan, China [44]). Other possible distributions are normal, used by commercial traffic micro-simulation software such as Aimsun [45] or Vissim [46], and also chosen in some scientific work (e.g. to characterize 1-day data from a bus stop in Chennai City, India [47]); Wakeby, which outperformed the log-normal distribution in a study with 3 months of data from 4 stops in Auckland, New Zealand [48]; or Erlang, proposed in a study that analyzed 435 records from 12 bus stops in Shanghai, China [49].

The subsystems that contribute to an IPTS often fail to properly capture information that would be useful for later analysis, because they usually have other goals: to support tactical planning and emergency response in the case of AVL, and to manage concessions for AFC. Consequently, a series of issues commonly arise, related to internal problems of each dataset or inconsistencies between them. Those within the scope of this work are [50]:

- Erroneous AFC records, which can be caused by malfunctions, atypical traveler behavior, emergency route detours or mishandling of the equipment by drivers and operators [2].
- Wrong AVL entries due to system failures, incorrect driver operations or termini-specific issues.
- Multiple records for the same AVL event, possibly with different attributes (timestamp, vehicle or route identification).
- Lost AVL or AFC events.
- Missing or wrong information to match passenger rides with materialized and planned trips.
- Uncertainty regarding whether a programmed trip actually took place.

In some cases, these problems can be so severe that researchers have developed methodologies that model public transport features indirectly, instead of using a more immediate, but error-prone alternative (e.g., utilizing AVL instead of AFC or automated passenger counter records to estimate public transport demand [51].

There are many published examples of the combined application of multiple automated collection data systems on the different aspects of urban transit management and planning. Among those that utilize AVL and AFC data, some noteworthy examples are:

- Space-temporal load profiles of urban transit vehicles during a month in The Hague (Netherlands), fully integrating GTFS records as a third data source with AVL and AFC check-in and check-out information [50].
- Offline processing of automated train tracking and magnetic trip card-based fare collection systems in San Francisco Bay Area (USA) [52].
- Estimation of origin-destination (OD) matrices and path choice models for rail passengers of the Chicago Transit Authority [4].
- Multi-modal trip purpose modeling and enhanced OD estimation in Queensland (Australia) [30].

- Metro and bus OD matrices, speed profiles of vehicles and quality service indicators, etc. for the Transantiago public transport system in Santiago de Chile [53].
- "Driver assisted bus interviews": if SC records are correctly linked with AVL information, they can function as revealed preference surveys [54].
- Tracking SCs along metro and bus to identify transfer behavior in Shenzhen (China), making use of bus AFC records that only show card id and sweeping time [55].

However, to the best of our knowledge there is room for improvement in the methodologies to apply in situations where AFC, AVL, and schedule data are available, but they are particularly challenging to fully make use of: information of varying reliability to differentiate trips within each of the 3 subsystems, but no direct way to identify entries of different subsystems that describe the same trip; missing of wrong AVL entries, AFC with wrong state information, or failing to correctly identify the current stop; only the planned (and sometimes, the detected) starts of the trips available from the scheduling subsystem, which may be stipulated at a stop 'downstream' the initial terminus of the route; users not requiring to check-out when leaving a vehicle; or unplanned trips that respond to daily operational decisions and are not shown in the schedule of the system. We are hopeful this work will be useful to other researchers and transportation engineers during their activities such as auditing, obtaining transit origin-destination matrices and travel patterns, user behavior modeling, or estimation of vehicle load profiles.

## III. METHODOLOGY

This section begins specifying the sources and expected structure of the input data. Then, it details the preprocessing steps that are applied to AVL, AFC, and planning information; representing each visit of a bus to a stop as a single event from each source. This is followed by the analysis of AVL data as sequences that are broken down in fragments of their respective routes, and the implementation of link travel times and dwell times distribution models. After that, the initial characterization of the performed trips takes place, which may be improved if necessary by detecting vehicle id changes mid-trip. Next, scheduled beginnings are linked to identify planned and extra trips, distinguishing in the first case if the intended vehicle was used or not; and to improve the fidelity of the recreation. Then, AFC events are assigned to bus calls. Finally, those trips supported by enough IPTS information will be accepted. Figure 1 shows an overall summary of the whole process.

### A. INPUT DATA

Table 1 contains a summary of the required bus stops, AFC, AVL, travel times lower bounds, schedule, and route-level data. It is worth noting that the ids of *bus stops*, *routes*, and *vehicles* need to be consistent throughout all the subsystems. The *group* columns in the AFC and AVL data should contain a unique identifier for each set of values from other columns present in their particular subsystems that can help to differentiate between runs of a vehicle.

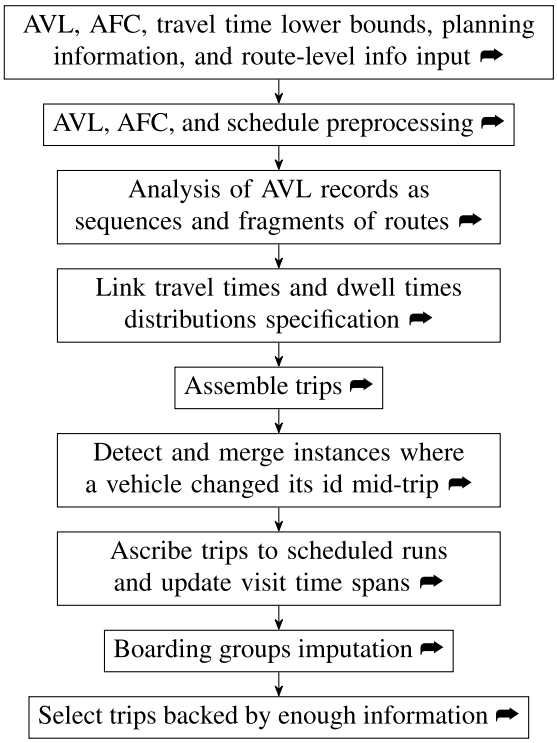



**FIGURE 1.** Methodology outline.

Regarding the schedule, the methodology is designed to work even when it is incomplete, or to detect unplanned trips. This section will assume that the three columns with temporal information may be available in at least part of the dataset.

#### 1) BUS STOPS

The location and name of the bus stops are needed. $\Xi$ is defined as a set composed by tuples $\xi_i$, which represent each of these entries, differentiated by a unique *id* $m_i$ (other variables not shown):

$$\begin{aligned} &\Xi = \{\xi_i = (m_i, \ldots)\} \\ &i: \text{ unique row id} \quad i \in \mathbb{Z} \\ &m: \text{ bus stop id (bus stops info)} \quad m \in \mathbb{Z} \end{aligned} \tag{1}$$

#### 2) TRAVEL TIMES LOWER BOUNDS

AVL events that imply impossible vehicle movements will be recognized and filtered out with a table of lower bounds for the travel times between stops.

#### 3) AFC

The methodology makes use of the ticketing system records: when did each transaction take place; which vehicle was boarded at which stop; and, if available, other non-temporal columns which can help to tell entries from different visits apart.

**TABLE 1.** Methodology input data.

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Bus stop id used throughout the IPTS. |
| location | (real, real) | Geographical coordinates. |
| name | text | Human-readable name. |

(a) *bus stops*

| *Column* | *Type* | *Description* |
|---|---|---|
| bus_stop_0 | integer | Initial stop id. |
| bus_stop_1 | integer | Final stop id. |
| free_flow_tt | time | Lower bound of the time a bus needs for this trip leg. |

(b) *travel times lower bounds*

| *Column* | *Type* | *Description* |
|---|---|---|
| bus_stop | integer | Bus stop id. |
| route | integer | Route id. |
| vehicle | integer | Vehicle id. |
| instant | timestamp | Validation time. |
| group | integer | UID for each set of values from other columns that differentiate distinct visits of the vehicle. |

(c) *raw AFC*

| *Column* | *Type* | *Description* |
|---|---|---|
| bus_stop | integer | Bus stop id. |
| route | integer | Route id. |
| vehicle | integer | Vehicle id. |
| stop_duration | time | How long the bus stayed. |
| instant | datetime | Arrival time to the station. |
| group | integer | UID for each set of values from other columns that differentiate distinct visits of the vehicle. |

(d) *raw AVL*

| *Column* | *Type* | *Description* |
|---|---|---|
| line | integer | Line id. |
| vehicle | integer | Vehicle id. |
| bus_stop | integer | Bus stop id. |
| planned_start | timestamp | Planned bus departure time. |
| recorded_arrival | timestamp | Detected bus arrival time. |
| recorded_start | timestamp | Detected bus departure time. |

(e) *raw schedule*

| *Column* | *Type* | *Description* |
|---|---|---|
| route | integer | Route id. |
| unreliabl_trm | boolean | True if undependable AFC and AVL at termini. |
| trp_strt_crrct | time | Mean span from AVL entries to trip start detection. |
| max_hdwy | time | Upper bound of the interval separating consecutive trips. |
| max_trp_lg | time | Travel time between consecutive stops upper bound. |
| min_rnd_trp | time | Lower bound of how long a round trip takes. |

(f) *route-level information*

#### 4) AVL

A registry of the visits of the buses to the stops is needed, including fields that provide temporal information, and that help differentiate the different runs of each offered route.

#### 5) SCHEDULE

This methodology utilizes the planned beginnings of trips along each route, characterized by which vehicle was going to be used, and where and when they start. Two other timestamps may be recorded by the IPTS, corresponding to the detected arrival and departure of the bus to the first stop of the new trip.

#### 6) ROUTE-LEVEL INFORMATION

Several aspects that describe each route as a whole are also used (eq. 2):

- Whether the timestamps of AVL and AFC events at the termini (locations where the problems described in the literature are usually more prominent) are particularly unreliable ($y$).
- If there is a systematic time deviation between the events recorded in the scheduling and AVL subsystems, the earlier can be corrected by the appropriate constant value $z$. This may happen for instance if the AVL stores when the doors of the buses close, while the scheduling subsystem registers the moment vehicles cross certain geofence.
- An upper bound of the headway between trips during normal operations, $s$.
- An upper bound $e$ of how long a trip leg connecting consecutive stops may last.
- A lower bound $d$ of how much time a vehicle needs to come back to a stop after traveling the whole route.

$$\begin{aligned} y &: \text{termini are unreliable} \quad \text{boolean} \\ z &: \text{trip start detection lag} \quad \text{time} \\ s &: \text{headway upper bound} \quad \text{time} \\ e &: \text{trip leg upper bound} \quad \text{time} \\ d &: \text{round trip lower bound} \quad \text{time} \end{aligned} \tag{2}$$

### B. PREPROCESSING

In this section new tables are created for the AFC (➦) and AVL (➦) datasets, synthesizing in a single entry the information that each raw source provides regarding a bus call. Also, lower travel time bounds are used to filter out unreliable AVL data.

Finally, the columns available from the scheduling subsystem for each planned trip beginning are analyzed, extracting the most specific arrival and departure times available; and a time buffer to search for its matching trip (➦).

#### 1) AFC

It is assumed that there are no duplicate rows in the raw AFC information, since due to the monetary repercussions of the data, ticketing information is managed in a very careful way. SC and manual payment operations are atomic: they are either completed successfully or do not happen.

As will be explained in detail in section III-E, AFC information (which provides one data point per validation) is used to deal with the limitations of the AVL data (ideally, one datum per bus visit). Thus, the objective is to classify as a single boarding event all the validations that happen each time

**TABLE 2.** Raw AFC preprocessing procedure. ***Vehicle***, ***route***, and ***group UID*** remain constant through this example.

| *stop id* | *rank over set* | *rank over subset* | *classif. variable* | *stop grp.* | *validation instant* | *time gap* | *below upper limit* | *intermed. AVL entry* | *board. grp. change* | *board. group id* |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ... | | | | | |
| A | 1 | 1 | 0 | A | 03-24 12:31:23 | <null> | | | 1 | 987 |
| B | 2 | 1 | 1 | B | 03-24 12:33:56 | <null> | | | 1 | 988 |
| C | 3 | 1 | 2 | C | 03-24 12:36:14 | <null> | | | 1 | 989 |
| C | 4 | 2 | 2 | | 03-24 12:36:16 | 00:00:02 | ✓ | ✓* | 0 | |
| D | 5 | 1 | 4 | D | 03-24 12:37:24 | <null> | | | 1 | 990 |
| E | 6 | 1 | 5 | E | 03-24 12:39:44 | <null> | | | 1 | 991 |
| E | 7 | 2 | 5 | | 03-24 12:45:22 | 00:05:38 | ✓ | ✓* | 0 | |
| F | 8 | 1 | 7 | F | 03-24 12:47:37 | <null> | | | 1 | 992 |
| G | 9 | 1 | 8 | G | 03-24 13:48:51 | <null> | | | 1 | 993 |
| G | 10 | 2 | 8 | | 03-24 13:50:59 | 00:02:08 | ✓ | ✓* | 0 | |
| G | 11 | 3 | 8 | | 03-24 13:53:04 | 00:02:05 | ✓ | ✓* | 0 | |
| G | 12 | 4 | 8 | | 03-24 14:11:11 | 00:18:07 | ✓ | ✗** | 1 | 994 |
| G | 13 | 5 | 8 | | 03-24 14:11:13 | 00:00:02 | ✓ | ✓* | 0 | |
| | | | | | ... | | | | | |
| B | 45 | 2 | 43 | B | 03-24 15:49:28 | <null> | | | 1 | 1004 |
| C | 46 | 3 | 43 | C | 03-24 15:51:33 | <null> | | | 1 | 1005 |
| D | 47 | 2 | 45 | D | 03-24 15:54:02 | <null> | | | 1 | 1006 |
| D | 47 | 2 | 45 | | 03-24 15:54:02 | 00:00:00 | ✓ | ✓* | 0 | |
| D | 48 | 3 | 45 | | 03-24 23:05:49 | 08:11:47 | ✗ | | 1 | 1007 |
| E | 49 | 2 | 47 | E | 03-24 23:09:09 | <null> | | | 1 | 1008 |
| | | | | | ... | | | | | |

*: No visit of this vehicle to a different bus stop was found in the AVL data between $t_i - \Delta t_i$ $(\omega_i)$ and $t_i$
**: The AVL data reveals that this vehicle visited bus stop K at 14:01:51

a bus calls at a stop. The first and last ticketing events of these ''boarding groups'' can be used as an approximation of when the bus arrived and left the stop. To create them, a three-part process is carried out:

- For each *vehicle*, *route*, and *group*; identify as a ''stop group'' each set of consecutive AFC events (➦).
- Some *stop groups* may contain payments or validations from unrelated events (for example two tap-ins of the same *stop group* may happen too far apart from each other, or the AVL data could have registered a call at another stop in between them). Two criteria are used to identify these instances, splitting *stop groups* in *boarding groups* (➦).
- Gather the results in the table *boarding_groups* (➦).

The rest of this section details and exemplifies each of these steps.

### *a: CREATE STOP GROUPS*

The AFC records pertaining each bus are analyzed, distinguishing *stop groups* of consecutive entries referring to the same *stop id*.

This procedure relies on the fact that, as is represented on table 2, for a set (i.c. the raw AFC entries linked to a single *vehicle*) where a relation (i.c. '*happened before*') can be used to establish rankings over the whole set (column 'rank over set') and also over the different subsets defined by a partition (i.c. entries with the same *vehicle*, *route*, *group*, and *bus stop id*; column 'rank over subset'), the difference between the rank over the whole set and over a particular subset (column 'classif. variable') provides a distinct value for the members of that subset that appear consecutively when ranking all elements of the whole set (i.c. the *stop group*, shown in column '*stop grp.*'). The meaning of the rest of the columns of table 2 and the coloring of the cells will be explained as it is mentioned thorough the rest of this description of the AFC preprocessing.

The process is explained as three consecutive tasks:

*Task 1: partition by vehicle, rank over each set*

AFC entries are grouped by *vehicle*, and then ranked chronologically, starting from the superset $\Omega$ that contains all entries from the *raw_afc* table:

$$\Omega = \{\omega_i = (t_i, v_i, a_i, \alpha_i, b_i, \ldots)\} \tag{3}$$

Each element $\omega_i$ is a tuple that represents one row of *raw_afc*:

$$\begin{aligned} &i : \text{unique row id} \quad i \in \mathbb{Z} \\ &t : \text{validation instant} \quad \text{full date (time)} \\ &v : \text{vehicle id} \quad v \in \mathbb{Z} \\ &a : \text{route id} \quad a \in \mathbb{Z} \\ &b : \text{bus stop id} \quad b \in \mathbb{Z} \\ &\alpha : \text{AFC group UID} \quad \alpha \in \mathbb{Z} \end{aligned} \tag{4}$$

Then, a partition $\Sigma$ of $\Omega$ is established, where each subset $X_{v_i}$ contains the entries from *raw_afc* of the bus $v_i$:

$$\Sigma = \left\{ X_{v_i} \subset \Omega \mid X_{v_i} = \left\{ (x_{v_i})_j \right\} = \{\omega_k \mid v_k = v_i\} \right\} \tag{5}$$

Equation (6) defines a binary relation $\Theta_{v_i}$ ('*happened after*') over each $X_{v_i}$ (also known as an endorelation):

$$\Theta_{v_i} = \left\{ \left( (x_{v_i})_l , (x_{v_i})_m \right) \mid t_l > t_m \right\} \tag{6}$$

$\Theta_{v_i}$ creates a total preorder over $X_{v_i}$, allowing to assign a rank $(\gamma_{v_i})_n$ to each of its elements (some may be tied with each other). For each $X_{v_i}$ there will be a set $\Gamma_{v_i}$ of ranks, as many as distinct timestamps:

$$\Gamma_{v_i} = \left\{ (\gamma_{v_i})_j \right\} = \left[1 \ldots \left|\Gamma_{v_i}\right|\right] \tag{7}$$

Finally, eq. (6) establishes the functions $\beta_{v_i}$ that link each element $(x_{v_i})_j$ of each subset $X_{v_i}$ to its rank within it. Thus, column 'rank over set' of table 2 contains the rank $\beta_{v_i}(\omega_i)$ of each *raw_afc* entry within the subset of all the rows related to vehicle $v_i$.

$$\beta_{v_i} : X_{v_i} \to \Gamma_{v_i}; \quad \forall (x_{v_i})_j, (x_{v_i})_k : \begin{cases} \beta_{v_i}\left((x_{v_i})_j\right) > \beta_{v_i}\left((x_{v_i})_k\right) \Leftrightarrow (x_{v_i})_j \ \Theta_{v_i} \ (x_{v_i})_k \\ \beta_{v_i}\left((x_{v_i})_j\right) < \beta_{v_i}\left((x_{v_i})_k\right) \Leftrightarrow (x_{v_i})_k \ \Theta_{v_i} \ (x_{v_i})_j \\ \beta_{v_i}\left((x_{v_i})_j\right) = \beta_{v_i}\left((x_{v_i})_k\right) \Leftrightarrow t_j = t_k \end{cases} \tag{8}$$

Task 2: *partition by vehicle, route, group, and bus stop; rank over each subset*

AFC entries are classified by *vehicle*, *route*, *group*, and *bus stop*; and ranked chronologically. These four columns of the *raw_afc* table remain constant during all the validations of a particular *boarding event*. The process is analogous to what has already been described in the first task. The family $\Phi$ partitions $\Omega$ in several $Y_{v_i,a_i,u_i,\alpha_i,b_i}$ subsets. Each one contains the entries from *raw_afc* that show in their columns *vehicle*, *route*, *group*, and *bus stop* the values defined by the tuple $(v_i, a_i, \alpha_i, b_i)$:

$$\begin{aligned} \Phi &= \left\{ Y_{v_i,a_i,\alpha_i,b_i} \subset X_{v_i} \,\middle|\, Y_{v_i,a_i,\alpha_i,b_i} \right. \\ &= \left\{ (y_{v_i,a_i,\alpha_i,b_i})_j \right\} \\ &= \left. \left\{ (x_{v_i})_k \mid a_k = a_i \wedge \alpha_k = \alpha_i \ \wedge \ b_k = b_i \right\} \right\} \end{aligned} \tag{9}$$

The binary relation $\Lambda_{v_i,a_i,\alpha_i,b_i}$ (again, '*happened after*') is characterized over each $Y_{v_i,a_i,\alpha_i,b_i}$ subset in eq. (10):

$$\Lambda_{v_i,a_i,\alpha_i,b_i} = \left\{ \left((y_{v_i,a_i,\alpha_i,b_i})_l, (y_{v_i,a_i,\alpha_i,b_i})_m\right) \mid t_l > t_m \right\} \tag{10}$$

Each element of $Y_{v_i,a_i,\alpha_i,b_i}$ can be mapped to a rank value $(\delta_{v_i,a_i,\alpha_i,b_i})_n$ thanks to the total preorder established by $\Lambda_{v_i,a_i,\alpha_i,b_i}$ over it. The set $\Delta_{v_i,a_i,\alpha_i,b_i}$ of all ranks of the elements of subset $Y_{v_i,a_i,\alpha_i,b_i}$ within it is:

$$\begin{aligned} \Delta_{v_i,a_i,\alpha_i,b_i} &= \left\{ (\delta_{v_i,a_i,\alpha_i,b_i})_j \right\} \\ &= \left[1 \ldots \left|\Delta_{v_i,a_i,\alpha_i,b_i}\right|\right] \end{aligned} \tag{11}$$

Lastly, (12) shows how functions $\varepsilon_{v_i,a_i,\alpha_i,b_i}$ link each element $(y_{v_i,a_i,\alpha_i,b_i})_j$ of each subset $Y_{v_i,a_i,\alpha_i,b_i}$ to its rank within it. Thus, column 'rank over subset' of table 2 contains the rank $\varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i)$ of each *raw_afc* entry within the subset of all the rows that share its *vehicle*, *route*, *group*, and *bus stop* values.

$$\varepsilon_{v_i,a_i,\alpha_i,b_i} : Y_{v_i,a_i,\alpha_i,b_i} \to \Delta_{v_i,a_i,\alpha_i,b_i}; \quad \forall \left( (y_{v_i,a_i,\alpha_i,b_i})_j, (y_{v_i,a_i,\alpha_i,b_i})_k \right) : \begin{cases} \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) \\ \quad > \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \\ \qquad \Leftrightarrow (y_{v_i,a_i,\alpha_i,b_i})_j \ \Lambda_{v_i,a_i,\alpha_i,b_i} \ (y_{v_i,a_i,\alpha_i,b_i})_k \\ \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) \\ \quad < \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \\ \qquad \Leftrightarrow (y_{v_i,a_i,\alpha_i,b_i})_k \ \Lambda_{v_i,a_i,\alpha_i,b_i} \ (y_{v_i,a_i,\alpha_i,b_i})_j \\ \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) \\ \quad = \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \Leftrightarrow t_j = t_k \end{cases} \tag{12}$$

Task 3: *create stop groups*

The difference between $\beta_{v_i}(\omega_i)$ from eq. (8) and $\varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i)$ returns the 'classif. parameter' of element $\omega_i$ ($\zeta(\omega_i)$, shown in table 2). This value, if one ranks chronologically all entries of set $X_{v_i}$ (the ones related to *bus* $v_i$), remains the same and is unique for each group of rows from its subset $Y_{v_i,a_i,\alpha_i,b_i}$ (those that report *bus*, *line*, *group*, and *bus stop* values of $v_i$, $a_i$, $\alpha_i$, and $b_i$) that appear consecutively.

$$\zeta(\omega_i) = \beta_{v_i}(\omega_i) - \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i); \quad \forall (\omega_i, \omega_j) : \begin{cases} v_i = v_j \ \wedge \ a_i = a_j \ \wedge \ \alpha_i = \alpha_j \ \wedge \ b_i = b_j \\ \quad \wedge \ \zeta(\omega_i) = \zeta(\omega_j) \Longleftrightarrow \omega_i \text{ and } \omega_j \text{ belong to} \\ \qquad\qquad \text{the same stop group.} \\ \text{Otherwise} \quad \Longleftrightarrow \omega_i \text{ and } \omega_j \text{ belong to} \\ \qquad\qquad \text{different stop groups.} \end{cases} \tag{13}$$

The column '*stop grp.*' of table 2 shows the outcome of this first approximation to the objective of identifying the *boarding groups*; displaying a single letter for all consecutive *raw_avl* entries with the same $v_i$, $a_i$, $\alpha_i$, $b_i$, and $\zeta(\omega_i)$ values. The coloring of columns 'stop id,' 'classif. variable,' and 'stop group' illustrates the classification process and its result. For instance, rows pertaining to calls to *bus stop* D are gathered in two *stop groups*, differentiated by $\zeta(\omega_i)$ values of 2 and 45.

A way to verify how this first task has performed is to study the 'time gap' (table 2) between consecutive rows of the same *stop group*, $\Delta t_i$ (eq. 14), noting that if $\omega_{i-1}$ does not belong to $Y_{v_i,a_i,\alpha_i,b_i}$, $\varepsilon_{v_i,a_i,\alpha_i,b_i}$ is not defined, so neither is $\Delta t_i$:

$$\Delta t_i = t_i - t_{i-1} \quad \text{if } \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_{i-1}) = \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i) - 1 \tag{14}$$

As this gap increases, it is more likely that the latter entry took place during a different visit of the bus (with no intermediate entries due to no validations being recorded until the bus came back). The next section studies this situation.

### *b: SPLIT STOP GROUPS IN BOARDING GROUPS*

The question regarding excessive time gaps between some of the entries that are part of the same *stop group*, is addressed with the following assumptions:

- In some cities, it is not uncommon for the driver to allow passengers to wait for the start of a trip inside the bus, especially if the weather is bad. However, if the separation between two consecutive entries of the same *stop group* is greater than the maximum headway $s$, their group will be split between them. This happens in the next to last row of table 2: the time elapsed since the previous validation is extremely long (represented with the symbol '✗' in column 'below upper limit'), so one can be certain that this row is describing a different *boarding event* and the group is split, as has been portrayed with the change in color from orange to red. An adequate value for this parameter will depend on the particularities of the case under analysis.
- For all other pairs of consecutive entries of the same *stop group*, if table *avl_coalesced* (defined later during the description of the pre-processing of the AVL data ➦) shows that the bus visited another stop between their *timestamps*, they belong to different *boarding groups*. An example of this situation can be found in the row with 'rank over set' = 12 of table 2, where the 5-entries *stop group* (G, 8) it is part of is split in two boarding groups (993 and 994); because, as the symbol ✗ of column '♯ AVL entry' denotes, between its timestamp (14:11:11) and the one from the previous entry (13:53:04) a lookup through the raw AVL data (not represented) has concluded that the bus called at bus stop K at 14:01:51.

These premises are utilized to define $\eta$, (eq. (15), column 'board. grp. change' of table 2), a value that will equal 1 if a row is the first of a boarding group, and 0 in other cases. $\sigma_j$ represents an entry of table *avl_coalesced* (➦), while $h_j$ and $p_j$ are its vehicle id and arrival time, respectively:

$$\eta(\omega_i) = \begin{cases} 0 & \text{if } \Delta t_i \leq s \\ & \quad \wedge\ \nexists \sigma_j \,\middle|\, h_j = v_i \\ & \qquad \wedge\ t_i - \Delta t_i \leq p_j \leq t_i \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

An index $\theta$ is then defined over $\Omega$, sorting its rows by *vehicle*, chronological *rank* within the entries of their vehicle (ascending), *bus stop* (ascending), and *boarding group change* (descending). The relative ordering of entries $\omega_i, \omega_j$ with $v_i = v_j$, $\beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j)$, $b_i = b_j$, and $\eta(\omega_i) = \eta(\omega_j) = 0$ is inconsequential. When ordered with this index, the elements of $\Omega$ will appear consecutively if they are part of a *boarding group*, with a value of *group change* of 1 for the first entry and 0 for the others up until its end. In other words, *group change* will equal one when a *stop group* commences (since a *boarding group* also will also begin) or when a *stop group* is split due to one of the two previous criteria (➦).

$$\begin{aligned} &\theta : \Omega \rightarrow \{1 \ldots |\Omega|\}; \\ &\theta(\omega_i) > \theta(\omega_j) \Longleftrightarrow v_i > v_j \\ &\qquad \vee\ v_i = v_j \wedge \beta_{v_i}(\omega_i) > \beta_{v_j}(\omega_j) \\ &\qquad \vee\ v_i = v_j \wedge \beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j) \wedge b_i > b_j \\ &\qquad \vee\ v_i = v_j \wedge \beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j) \wedge b_i = b_j \\ &\qquad\qquad \wedge\ \eta(\omega_i) \leq \eta(\omega_j) \end{aligned} \tag{16}$$

The *boarding group id*, $o(\omega_i)$, of each row $\omega_i$ is the sum of all $\eta(\omega_j)$ values from rows $\omega_j$ such that $\theta(\omega_j) \leq \theta(\omega_i)$:

$$o(\omega_i) = \sum_{j|\theta(\omega_j) \leq \theta(\omega_i)} \eta(\omega_j) \tag{17}$$

Going back to Table 2, the content and colors of the cells of columns 'time gap,' 'below upper limit,' '♯ intermediate AVL entry,' 'boarding group change,' and 'boarding group id' have been chosen to describe how stop groups are split in boarding groups:

- If an entry is the first of its *stop group* ('time gap' = ⟨null⟩), a new *boarding group* should also begin (rows with 'rank over set' ∈ {1, 2, 3, 5, 6, 8, 9, 45, 46, 47, 49}). 'group change' equals 1, and there is no need to check columns 'below upper limit' or '♯ intermediate AVL entry.' For each of these rows, the columns involved in the identification of their *stop group* and *boarding group* are filled with the same color, different from their respective predecessors.
- If the lapse between two successive validations of the same *stop group* is too long, they are the end and beginning of two different *boarding groups*. The latter row shows the symbol ✗ at 'below upper limit,' while its column '♯ intermediate AVL entry' is not needed, and 'group change' is 1. It also depicts its whole decision process, utilizing one color for '*stop id*,' 'grouping parameter,' and 'stop group'; and another for 'below upper limit' and 'boarding group id,' showing how each *stop group* is split in *boarding groups*.
- For the remaining pairs of consecutive rows that share the same *stop group*, the symbol of column '♯ intermediate AVL entry' will indicate whether they belong to the same *boarding group*:
  - ✗: The vehicle related to both entries has moved to another stop (and eventually back) at a time between their timestamps, so they belong to different *boarding groups*. Again, 'group change' = 1, and colors illustrate the reasoning behind this decision: one color for '*stop group*' and the first *boarding group,* and a different one for the second *boarding group* created by the split.
  - ✓: There is no evidence that the vehicle has moved between the timestamps of both entries, so it is concluded that they belong to the same *boarding group*: 'group change' = 0. The columns of the

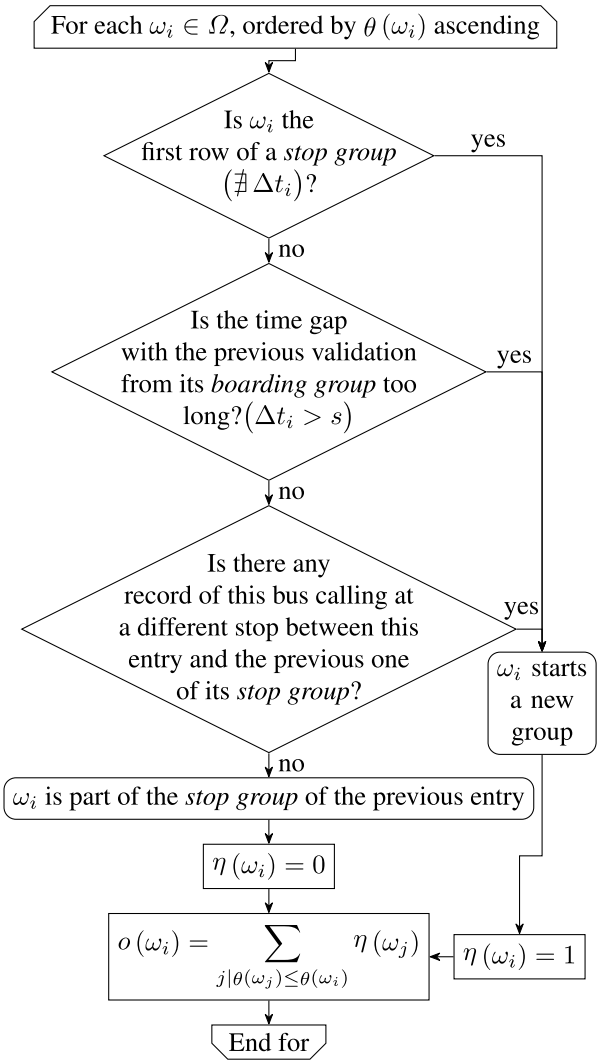



**FIGURE 2.** Splitting stop groups in boarding groups.

latter row that decide its *stop group* and *boarding group* show the same colors as in the former.

Alternatively, fig. 2 illustrates how stop groups are separated in boarding groups with a flow diagram.

Finally, for each boarding group $o_x$, the instants of its first $\vartheta(o_x)$ and last $\iota(o_x)$ validations are computed, as well as how long it lasted $\kappa(o_x)$:

$$\begin{aligned} \vartheta(o_x) &= \min\left(\{t_i \mid o(\omega_i) = o_x\}\right) \\ \iota(o_x) &= \max\left(\{t_i \mid o(\omega_i) = o_x\}\right) \\ \kappa(o_x) &= \iota(o_x) - \vartheta(o_x) \end{aligned} \tag{18}$$

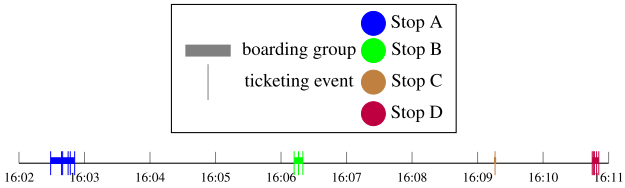
*Boarding groups* that last longer than the maximum headway for their route ($s$) will be considered to originate from unreliable data and won't be utilized to infer missing visits to stops not recorded by the AVL.

### c: OUTPUT

The results of the AFC pre-processing are gathered in the table *boarding_groups*, structured as shown in table 3, while

**TABLE 3.** AFC pre-processing output: ***boarding_groups***.

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Id of the boarding group. |
| bus_stop | integer | Bus stop id. |
| vehicle | integer | Vehicle id. |
| route | integer | Route id. |
| group | integer | AFC group UID. |
| boarding_range | time range | [earliest validation, latest validat.] |




AFC data sample from one vehicle

**FIGURE 3.** Transition from individual ticketing events to encompassing ***boarding groups***.

fig. 3 depicts an example transition from 15 individual ticketing events to 4 encompassing *boarding groups* at different stops.

### 2) AVL

The procedures detailed in this section aim to characterize the movement of the vehicles with a single record for each stop of each trip.

#### a: FILTER OUT DUPLICATE ROWS

The first action is to identify and filter out duplicate entries, creating the multiset $\Theta$ of all *raw_avl* rows, defined over the set $\Lambda$ of distinct AVL records $\lambda_i$; and the multiplicity function $\zeta$ that returns how many times each $\lambda_i$ appears in the raw AVL dataset:

$$\begin{aligned} \Theta &= \langle \Lambda, \zeta \rangle = \left\{ \vartheta_i = {\lambda_i}^{\zeta(\lambda_i)} \right\} \\ \Lambda &= \left\{ \lambda_i = (t_i, v_i, a_i, b_i, \beta_i, \vartheta_i) \right\} \\ \zeta &: \Lambda \to \mathbb{Z}_{\geq 1} \end{aligned} \tag{19}$$

Each $\lambda_i$ is a tuple with the fields described in eq. (20):

$$\begin{aligned} &i : \text{unique row id} \quad i \in \mathbb{Z} \\ &t : \text{instant} \quad \text{full date (time)} \\ &v : \text{vehicle id} \quad v \in \mathbb{Z} \\ &a : \text{route id} \quad a \in \mathbb{Z} \\ &b : \text{bus stop id} \quad b \in \mathbb{Z} \\ &\beta : \text{AVL group id} \quad \beta \in \mathbb{Z} \\ &\vartheta : \text{stop duration} \quad \text{time} \end{aligned} \tag{20}$$

#### b: REMOVE ROWS NOT LINKED TO A REAL BUS STOP

It is assumed that entries with a *bus stop id* not found in set $\Xi$ (defined in eq. 1) are caused by exceptional events that do not happen consistently as the buses travel their routes, and are not linked to a position change. Equation (21) establishes $\Psi$, a subset of $\Lambda$ after these have been filtered out:

$$\Psi = \left\{ \psi_i = \lambda_j \mid \exists \xi_k : b_j = m_k \right\} \tag{21}$$

**TABLE 4.** How entries of a trajectory linked to a single visit are identified.

| *Stop* | *Chronological rank over trajectory* | - | *Chronological rank over (trajectory, stop)* | = | *Visit group number* | *Stop* |
|---|---|---|---|---|---|---|
| A | 1 | - | 1 | = | 0 | A |
| B | 2 | - | 1 | = | 1 | B |
| C | 3 | - | 1 | = | 2 | C |
| C | 4 | - | 2 | = | 2 | |
| D | 5 | - | 1 | = | 4 | D |
| E | 6 | - | 1 | = | 5 | E |
| C | 7 | - | 3 | = | 4 | C |
| F | 8 | - | 1 | = | 7 | F |
| | | | ... | | | |

**TABLE 5.** Treatment of multiple AVL entries from the same visit to a stop.

| *instant* | *duration* | *stop* |
|---|---|---|
| | ... | |
| 12:50:29 | 0 | 151 |
| 12:51:11 | 19 | 135 |
| 12:51:18 | 198 | 135 |
| 12:51:47 | <null> | 135 |
| 12:55:07 | 14 | 134 |
| | ... | |

(a) *avl*

| *arrival* | *departure* | *stop* |
|---|---|---|
| | ... | |
| 12:50:29 | 12:50:29 | 151 |
| 12:51:11 | 12:54:36 | 135 |
| 12:55:07 | 12:55:21 | 134 |
| | ... | |

(b) *avl_coalesced*

### c: IDENTIFY TRAJECTORIES

The next step is to utilize the columns from the AVL data to differentiate between the runs that constitute the public transportation supply. To this end, in this work a '*trajectory*' is defined as consecutive AVL records that share the same *vehicle*, *route*, and *group*. The relation $R$ maps each element of $\Psi$ with distinct values of $h, f$, and $\beta$ to a different *trajectory id* $(r)$:

$$R(\psi_i) = r_i \in \mathbb{Z}; \quad \forall (\psi_i, \psi_j): \begin{cases} r_i = r_j \Leftrightarrow a_i = a_j \ \wedge \ v_i = v_j \ \wedge \ \beta_i = \beta_j \\ \qquad \Leftrightarrow \psi_i \text{ and } \psi_j \text{ belong to the same trajectory.} \end{cases} \tag{22}$$

### d: DETERMINE VISIT GROUPS

Table 4 depicts how each trajectory is examined to tell apart those occasions when more than one row is added to the dataset for the same call at a stop (for example, when the doors are re-opened to let a late passenger in the bus). The procedure to identify these '*visit groups*' (calculate each entry's ranks over its trajectory, and among those records with the same trajectory and stop values; and then evaluate each element's classification variable as their subtraction) is similar to the one that has already been described and implemented in page 128255 to find *stop groups* in the AFC data. Its result is a relation $M$ (eq. (23)) which assigns the same *visit group* id $(\mu_i)$ to consecutive entries of a *trajectory* that happen in the same *bus stop*:

$$M(\psi_i) = \mu_i \in \mathbb{Z}; \quad \forall (\psi_i, \psi_j): \begin{cases} \mu_i = \mu_j \Longleftrightarrow \psi_i \text{ and } \psi_j \text{ are part of the same visit group.} \\ \mu_i \neq \mu_j \Longleftrightarrow \psi_i \text{ and } \psi_j \text{ are part of different visit groups.} \end{cases} \tag{23}$$

### e: MERGE ENTRIES OF EACH VISIT GROUP

The set $\Sigma$ summarizes the information pertaining the *visit groups*. Its elements contain the fields shown in eq. (24), and are stored in the table *avl_coalesced* (table 6b):

$$\Sigma = \{\sigma_{\mu_i} = (r_{\mu_i}, b_{\mu_i}, n_{\mu_i}, p_{\mu_i})\} \tag{24}$$

Each of its elements $\sigma_{\mu_i}$ is a tuple with the characteristics of the *visit group* $\mu_i$ (table 5 outlines this process):

- Its unique id *visit group id* $(\mu_i)$.
- Its *trajectory* $r_{\mu_i}$: The characteristics that define it (*route*, *vehicle*, and *group UID*) will be referred to as $r_{\mu_i}$, $v_{\mu_i}$, and $\beta_{\mu_i}$.
- Its *bus stop* $b_{\mu_i}$.
- The moment $n_{\mu_i}$ the bus arrived at the stop, defined as the minimum instant $(w)$ from all elements of $\Psi$ that are part of this *visit group*:

$$n_{\mu_i} = \min\left(\{w_j \mid \psi_j : M(\psi_j) = \mu_i\}\right) \tag{25}$$

- The instant $p_{\mu_i}$ when the bus left the stop defined as the maximum of these two values:
  - $(p_{\mu_i})_1$: The maximum of the addition of the *instant* $(w_j)$ and the duration $(\vartheta_j)$ for those elements $\psi_j$ of the *visit group* where *stop duration* is defined.
  - $(p_{\mu_i})_2$: The maximum of the *instant* $(w_j)$ for those elements $\psi_k$ of the *visit group* that do not report a *stop duration* value $(p_k = \langle\text{null}\rangle)$.

$$\begin{aligned} (p_{\mu_i})_1 &= \max\left(\{w_j + \vartheta_j \mid \psi_j : M(\psi_j) = \mu_i \wedge \exists p_j\}\right) \\ (p_{\mu_i})_2 &= \max\left(\{w_k \mid \psi_k : M(\psi_k) = \mu_i \wedge \nexists p_j\}\right) \\ p_{\mu_i} &= \max\left((p_{\mu_i})_1, (p_{\mu_i})_2\right) \end{aligned} \tag{26}$$

### f: IDENTIFY AND REMOVE UNFEASIBLE OR UNREALISTIC TRIP LEGS

Regarding each trajectory as a series of trip legs between its *visit groups*, those shorter than the free flow time between the involved stops are not possible. Two possible situations arise:

- Moving backwards the departure time in the former stop, thus increasing the leg length, solves the issue. This amounts to assuming that the information regarding how long the bus stayed in the initial stop of the leg is not reliable.
- Not even setting the dwell time in the former stop to zero leaves enough time to travel to the latter. In this case, both *visit groups* will be considered as unreliable and removed.

Also, those trip legs longer than the upper bound $e$ for their route will be used to split their trajectories. Thus, AVL entries that present the same *vehicle*, *route*, and *group*, but separated by a trip leg too long to have occurred during a single trip, will be considered separately.

### g: OUTPUT

Table 6 shows how the outcome of AVL preprocessing is stored in tables *trajectories* and *avl_coalesced*.

**TABLE 6.** AVL preprocessing output.

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Id of the visit. |
| route | integer | Route id. |
| vehicle | integer | Vehicle id. |
| group | integer | AVL group UID. |
| stops_sequence | integer | Stops seq. id (described later). |
| trajectory_range | time range | [traject. start, traject. end] |

(a) *trajectories*

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Id of the visit. |
| bus_stop | integer | Bus stop id. |
| ord_wthin_trj | integer | Chronological order within traject. |
| trajectory | integer | Trajectory id. |
| avl_range | time range | [arrival, departure] |

(b) *avl_coalesced*

**TABLE 7.** Raw schedule preprocessing output: *schedule*.

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Planned start id. |
| bus_stop | integer | Stop id. |
| vehicle | integer | Vehicle id. |
| trp_srch_buff | t. range | [t. min, t. max] to match to a trip. |
| sched_range | t. range | [arrival, depart.] from the sched. subsystem. |

### 3) SCHEDULE

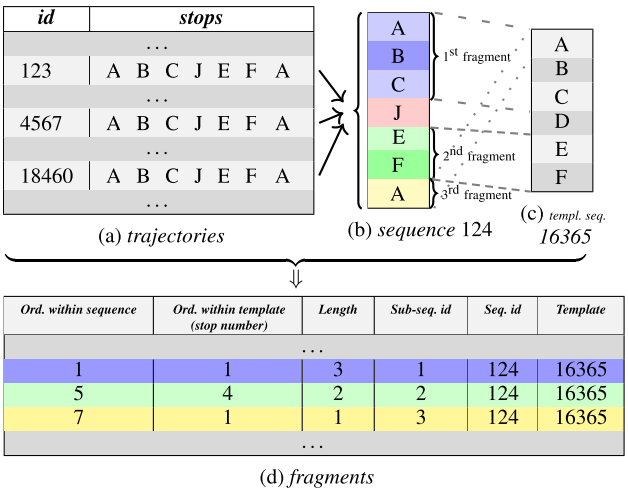
Firstly, the events recorded in the scheduling subsystem should be corrected by the appropriate value $z$, if defined for the corresponding route.

Then, a time range $n$ is created for each planned trip, encompassing the arrival and departure times that can be deducted from the most specific available columns, as long as they provide coherent information (e.g.: departures cannot happen before arrivals). Another time buffer $q$ is also created around its planned *start time* $t_p$, with a semi-width equal to the maximum headway $s$. It will be used in section III-G to match each entry of the schedule to the trip that materializes it. Equations (27) and (28) respectively enunciate the parameter and variables, and detail the conditions just described; while table 7 shows the structure of the planning information after preprocessing.

$$\begin{aligned} &t_p : \text{planned depart. t.} \quad \text{full date (time)} \\ &t_d : \text{recorded depart. t.} \quad \text{full date (time)} \\ &t_a : \text{recorded arriv. t.} \quad \text{full date (time)} \\ &n : \text{visit range from sched. subsystem} \\ &\qquad \left[\text{arrival time, departure time}\right] \\ &q : \text{trip search buffer} \quad \left[\text{lower time bound, upper time bound}\right] \end{aligned} \tag{27}$$

$$n = \begin{cases} [t_a, t_d] & \text{if } t_a \le t_d \\ [t_p, t_d] & \text{if } (t_a > t_d \vee \nexists t_a) \wedge t_p \le t_d \\ [t_d, t_d] & \text{if } (t_a > t_d \vee \nexists t_a) \wedge t_p > t_d \\ \langle null \rangle & \text{otherwise} \end{cases}$$
$$q = [t_d - s, t_d + s] \tag{28}$$

**TABLE 8.** Analysis of the trajectories of a route for one of its template sequences.




| *Ord. within sequence* | *Ord. within template (stop number)* | *Length* | *Sub-seq. id* | *Seq. id* | *Template* |
|---|---|---|---|---|---|
| 1 | 1 | 3 | 1 | 124 | 16365 |
| 5 | 4 | 2 | 2 | 124 | 16365 |
| 7 | 1 | 1 | 3 | 124 | 16365 |

(d) *fragments*

**TABLE 9.** AVL sequences analysis output.

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Id of the sequence. |
| route | integer | Line id. |
| stops_sequence | tuple of int | Sequence of stop ids. |

(a) *stops_sequences*

| *Column* | *Type* | *Description* |
|---|---|---|
| id | integer | Sequence id. |
| n_stops | integer | Number of stops (materialized for convenience). |
| name | text | Human-friendly name. |

(b) *template_sequences*

| *Column* | *Type* | *Description* |
|---|---|---|
| ord_within_seq | int | Event ordinal within its sequence |
| stop_number | int | Event ordinal within the template. |
| fragment | int | fragment id within its sequence. |
| sequence | int | sequence id. |
| template | int | template sequence id. |

(c) *fragments*

## C. ANALYZE AVL TRAJECTORIES AS SEQUENCES AND FRAGMENTS OF ROUTES

AVL trajectories are analyzed as just ordered *sequences* of stops, which will be the building blocks to assemble the full trips that have occurred, defined by their "template sequences." Table 8 illustrates this process, and table 9 gathers the outputs of its three steps:

### 1) IDENTIFY DISTINCT AVL TRAJECTORY SEQUENCES

An id is assigned to each unique stops sequence extracted from the trajectories of each route, as shown in tables 8a, 8b and 9a. The field *stops_sequence* of the trajectories table (➦) signifies this relation.

### 2) SINGLE OUT TEMPLATE SEQUENCES

This methodology assumes that each route can be split in a series of "subroutes" that represent the trips that compose it (for instance, the trips back and forth between the termini of a linear route; or a single round-trip in the case of circular routes). Each subroute is characterized by its "template sequence" of stops table 8c) that a typical, completely carried

out, perfectly recorded run of that *subroute* must follow. They can be known beforehand, or ascertained through the examination of the sequences of stops found during their previous step and their relative frequencies, since the templates will very likely be among those found most often. They are stored as illustrated in tables 9b.

The elements of each of the *template sequences* of each route can be uniquely identified by their ordinal (the ''stop number'').

#### 3) BREAK DOWN SEQUENCES IN TEMPLATE FRAGMENTS

As depicted in tables 8b to 8d, the sequences followed by the trajectories can be split in:

- Continuous *fragments* of their route's *template sequences* (i.e., no elements missing between their extremes), that represent parts of trips that the AVL system managed to record correctly. They allow to view each trajectory found in the AVL data as a series of *segments* that fit in its template. table 9c how they are stored.
- Incompatible portions (caused by erroneous entries in the AVL subsystem; the vehicle carrying out other *subroute*; or incorrect operations, e.g., not updating the on-board computer to reflect that the bus is following a different route).

### D. CHOOSE LINK TRAVEL TIMES AND DWELL TIMES DISTRIBUTION MODELS

These models are utilized as part of the criteria for identifying AVL fragments or boarding groups that are part of the same trip; to infer missing stop information; and to filter out erroneous recorded trip starting times. For each route, link travel times between consecutive stops, and dwell times for all of them but the last one, will be needed.

They should consider known factors that modify travel and dwell times in the area of study, such as the time, whether it is a working day or not, or seasonal mobility changes.

### E. ASSEMBLE TRIPS

Trips are constructed starting from a ''seed'' that is completed backwards and forward in time, looking for AVL segments and *boarding groups* events part of the same subroute and with the same *vehicle id* as the seed that, according to the instant of the furthermost known data point in the current growth direction and the probability distributions of the duration of unknown intermediate trip legs and calls at stops, fall within their **minimum-amplitude prediction interval of probability *g***.

'***g***' is a parameter of this methodology (eq. 29). The closer it is to one, the wider and more computer-intensive the search needs to be, and the risk of considering invalid or unrelated events as part of the current trips increases. If set too low however, events that really were part of the trip that is being characterized may be ignored.

$$\boldsymbol{g} : \text{probability of the prediction intervals} \quad \boldsymbol{g} \in [0, 1] \tag{29}$$

For each subroute and direction (backwards or forward in time), the seeds are selected following two consecutive iterative process. Firstly, by looping over the AVL fragments with a length of at least ***c***, from longer to shorter. '***c***' is the parameter '***minimum AVL seed length***' (eq. 30). This decision stems from the hypothesis that **longer AVL trajectory fragments are more likely to be reliable**, while shorter ones may be caused by clock, GPS, or operation errors. After that, those *boarding groups* not filtered out will be also used as seeds. The algorithm will skip those seeds contained in the tables of events to be ignored (explained later ➡).

$$\boldsymbol{c} : \text{min. AVL seed length} \quad \boldsymbol{c} \in \mathbb{N} - \{0\} \tag{30}$$

Once a seed has been established, it ''grows'' both back and forward in time, following a procedure that bears similarity to dead reckoning: starting from the furthest known point in a direction (the initial fix), minimum-amplitude prediction intervals of probability ***g*** for the departures or arrivals (if traversing backwards or forward, respectively) of the calls to consecutively farther away stops are computed as the sum of the involved travel and dwell times from intermediate stops, until one of following conditions is reached (checked in this order) and a new fix is selected:

- The prediction interval intersects the *avl_range* of at least a record from table *avl_coalesced*. In this case, the closest to the most likely arrival and departure times range is chosen, and a portion of its encompassing fragment is identified and added to growing new trip, from said record up to what comes first between:
  - The next-to-last or second stop of the route, while growing forward or backwards, respectively.
  - The end of its fragment in the current growth direction.

  This distinction aims to on one hand to save computer time, by adding in a single step several calls of the vehicle; and it also makes sure that a feasibility range is always calculated at the termini. Besides being used as part of the current process, to filter out unrealistic IPTS entries at those stops; they will be used to decide the best way to include the information available from the schedule.
- The prediction interval intersects the *boarding_range* of at least a compatible *boarding group*. The closest to the most likely arrival and departure times range is chosen.
- If the stop under scrutiny is a terminus, the most likely arrival (or departure, if growing the trip backwards) and dwell time are chosen.

In the first or second conditions, ''compatible'' means that it refers to the same route and vehicle as the seed; and is not in the table of events to be ignored (explained later ➡). Also, if more than one possibility appears, the most likely one according to the utilized link travel time and dwell time distributions is selected.

In all three instances, once the new fix has been selected, the set of most likely values for link travel times and dwell times will be used to infer the arrival and departure times at missing intermediate stops.

After reaching a terminus, growth in the current direction ends. For those routes where data at the termini have been deemed particularly unreliable ($y$ = true), if the call at the closest stop is backed by AVL or AFC data, arrival and departure times will always be inferred.

Once a seed has grown to encompass a full trip, as described by its *template*; a buffer encompassing it is created, extending backwards and forwards in time from each call's respective arrival and departure, adding the round-trip time lower bound for the corresponding route ($d$). AVL segments and *boarding groups* that overlap it are added to the tables of elements to ignore during the reminder of the trip assembly process. This procedure serves two purposes: to enforce that no event is utilized as part of more than one trip, and that vehicles follow feasible itineraries (enough time passes before they return to the same stop, as part of another trip).

Figure 4 displays a flowchart of the first part of this process, which utilizes segments of AVL data as seeds. The second part is completely analogous, but for the fact that only the remaining AFC information is utilized.

Figure 5 shows a complete example. Its main steps are:

1. The initial seed is an AVL segment that goes from the arrival at :20:13 at AB, to the departure at :22:41 from AE.
2. It grows backwards, utilizing the search range [:18:51, :19:53] at the terminus AA. It has been defined setting the arrival at AB as a fixed point, and calculating the prediction interval of probability $\boldsymbol{g}$ for the presence of the vehicle at AA.
   - A single compatible overlapping AVL event is found (3a), with arrival and departure times :18:31 and :18:55, respectively:
     - If the readings at the termini for this route have been deemed as reliable as in other stops ($y$ = false), 3a will be accepted as the call of the bus at the initial terminus.
     - Otherwise, since the fix for the search is in the stop next to the terminus (3c), the inferred visit 3b, from :19:15 to :19:46, will be preferred.

   Since this is one of the route's termini, the growth backwards ends.
4. Growing forward, the search range to be used at AF is computed, utilizing as a fixed reference the departure time from AE (:22:41). The result is the prediction interval of probability $\boldsymbol{g}$ of the presence of the bus at AF: [:23:04, :24:05], which intersects no compatible entry from the AVL or AFC subsystems.
5. The script keeps searching forward. At AG, another prediction interval of probability $\boldsymbol{g}$ is created for the arrival of the bus. This time, the sum of the individual distributions of travel times from AE to AF, and from AF to AG; and of the dwell time at AF will be needed.

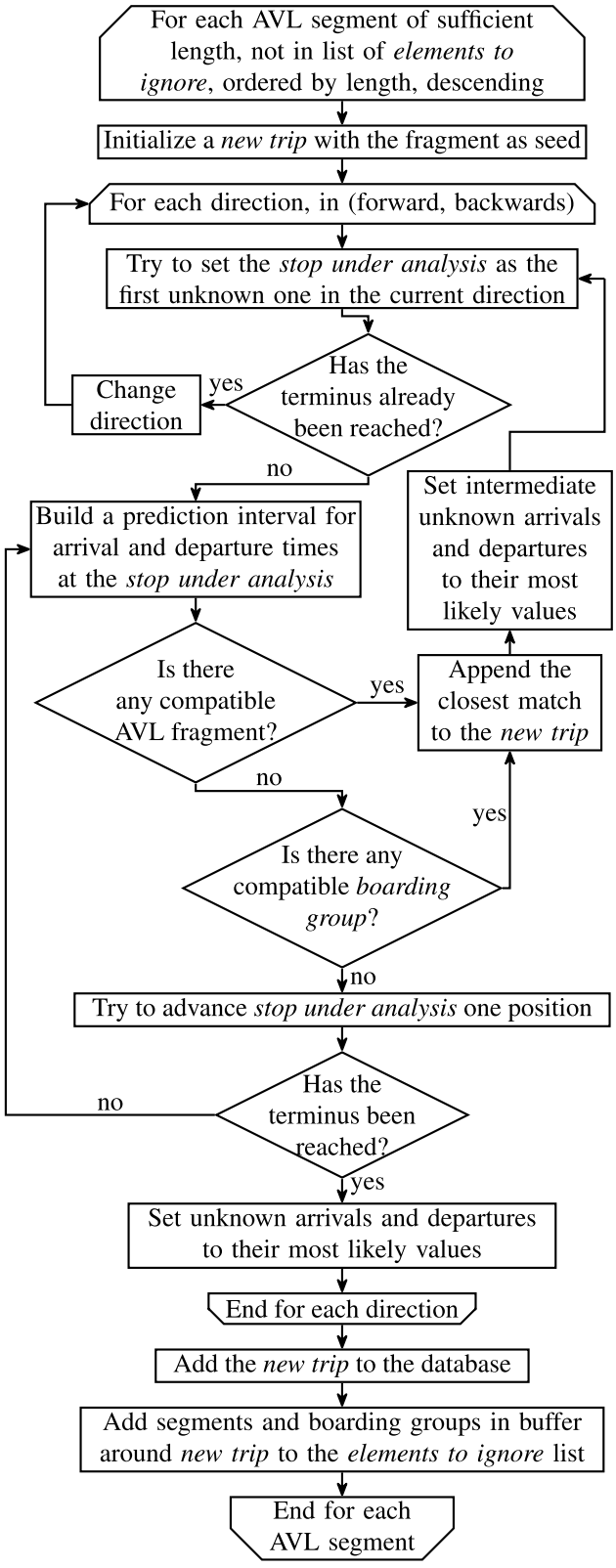



**FIGURE 4.** Trip inference process from AVL seeds.

**TABLE 10.** Trips characterization.

| *Column* | *Type* | *Description* |
|---|---|---|
| vehicle | int | Vehicle id. |
| id | int | Trip id. |
| merged_trip | int | Encompassing trip. id (if appl.). |
| template_sequence | int | Template sequence id. |
| scheduled_beginning | int | Scheduled beginning id (if appl.). |
| trip_range | t. range | [1st stop depart., last stop arriv.] |
| merged_trips | int tuple | Encompassed trips ids (if appl.). |

(a) *trips*

| *Column* | *Type* | *Description* |
|---|---|---|
| stop_number | int | Ordinal of stop in template. |
| id | int | Id of the visit. |
| trip | int | Trip the visit is part of. |
| avl_coalesced_id | int | *avl_coalesced* source (if appl.). |
| boarding_group_id | int | *boarding_group_id* source (if appl.). |
| visit_range | t. range | [arrival, departure] |

(b) *visits*_to_stops

| *Column* | *Type* | *Description* |
|---|---|---|
| origin | int | Ordinal of the last stop with IPTS data. |
| stop_number | int | Ordinal of the stop under scrutiny. |
| trip | int | Trip the visit is part of. |
| search_range | t. range | Bounds of the prediction interval. |

(c) *search_ranges*

The ensuing range ([:23:17, :24:57]) overlaps with a *boarding group* ([:23:55, :23:59]). Its earliest and latest ticketing events will be used as an approximation of the arrival and departure at AG.

6. Considering now the gap of 1m14s between the bus leaving AE at :22:41 and arriving at AG at :23:55; the most likely combination of the travel times from AE to AF and from AF to AG; and of dwell time at AF that add up to it is, according to their respective probabilistic distributions, 45s, 26s, and 3s; respectively. Thus, arrival and departure times at AF are set to [:23:26, :23:29].
7. Again, the search takes place at stop H, finding a compatible AVL entry. This one, and other three from the same fragment are added to the trip.
8. Several intermediate stops had to be inferred between the departure from AK and the arrival and AR. Missing arrival and departure times will be set to their most probable values, according to the 7 travel time and 6 dwell time distributions involved.
9. Finally, the other terminus of the route is reached. Since no compatible AVL or AFC is found, the arrival at this stop; as well as arrivals and departures at others downstream the last known departure, if any; are set to their mean values.

The output after all AVL and AFC data have been utilized (table 10) is the result of this methodology, and consists of three tables:

- *trips*, which synthesizes each of the trips that have been detected by this methodology.
- *visits_to_stops*, that characterizes each trip.
- *search_ranges*, where the prediction intervals utilized during the creation of the trips are saved.

### F. (OPTIONAL) DETECT AND MERGE INSTANCES WHERE A VEHICLE CHANGED ITS ID MID-TRIP

Due to the way operations are handled by the IPTS, some vehicles may change their *id* mid-trip, as it happens in the case study analyzed in this paper. They can be detected in this methodology as two extremely close in time "former" and "latter" trips, where a single vehicle could have provided all non-inferred *visits_to_stops*. This section follows the nomenclature described in (31).

$$\begin{aligned}
\phi, \lambda &: \text{formr, lattr } \textit{trip ids} \quad \phi, \lambda \in \mathbb{Z} \\
C &: \text{"should be merged" relation} \\
&\quad C = \left\{ (\phi, \lambda) \mid \phi, \lambda \text{ are the same trip} \right\} \\
T &: \text{instants} \quad \text{possible full dates (time)} \\
R &: \text{time ranges} \quad R = \left\{ (\rho[0], \rho[1]) \in T^2 \mid \rho[0] \leq \rho[1] \right\} \\
\&\& &: \text{"overlap" relation} \quad \&\& = \left\{ (\rho_i, \rho_j) \mid \right. \\
&\quad \rho_j[0] \leq \rho_i[1] \leq \rho_j[1] \\
&\quad \vee \rho_j[0] \leq \rho_i[0] \leq \rho_j[1] \\
&\quad \left. \vee \rho_i[0] < \rho_j[0] \wedge \rho_i[1] > \rho_j[1] \right\} \\
\sigma_i &: \text{trip } i \text{ range} \quad \sigma = (\text{start, end}) \in R \\
\upsilon_{i,j} &: \text{visit range for trip } i \text{ at stop } j \\
&\quad \upsilon = (\text{arrivl, depart.}) \in R \\
\epsilon_{i,j} &: \text{search range for trip } i \text{ at stop } j \\
&\quad \epsilon = (\text{lower bound, upper bound}) \in R \\
\tau &: \text{stop number} \quad \tau \in \mathbb{N} \\
\mu_i &: \text{lower bound of travel t. from stop } i \\
&\quad \text{to } i+1 \quad \text{time}
\end{aligned} \tag{31}$$

To correct this problem, the authors propose the following procedure to be carried out for each *template sequence*:

#### 1) IDENTIFY PAIRS OF TRIPS THAT SHOULD BE COMBINED

- To save computational time, only those trips that present overlap between their corresponding *trip_ranges* will be considered:

$$\phi \; C \; \lambda \implies \sigma_\phi \; \&\& \; \sigma_\lambda \tag{32}$$

- Also, for each call of each trip to a stop, a time buffer is created, as the smallest one that includes its *visit_range* and, if exists, its *search_range*. Two trips happen closely enough to be viable candidates when their time buffers overlap.

$$\begin{aligned}
\phi \; C \; \lambda \Rightarrow \exists \tau \mid \upsilon_{\phi,\tau} \; \&\& \; \upsilon_{\lambda,\tau} \vee \epsilon_{\phi,\tau} \; \&\& \; \upsilon_{\lambda,\tau} \\
\vee \; \upsilon_{\phi,\tau} \; \&\& \; \epsilon_{\lambda,\tau} \vee \epsilon_{\phi,\tau} \; \&\& \; \epsilon_{\lambda,\tau}
\end{aligned} \tag{33}$$

- Finally it must be possible, taking into account the lower bounds of the travel times between stops, for a single bus to perform all *visits_to_stops* entries from both trips that stem from the IPTS data. How this condition is met depends on the highest *stop_number* for which the "former" trip presents a non-inferred *visits_to_stops*

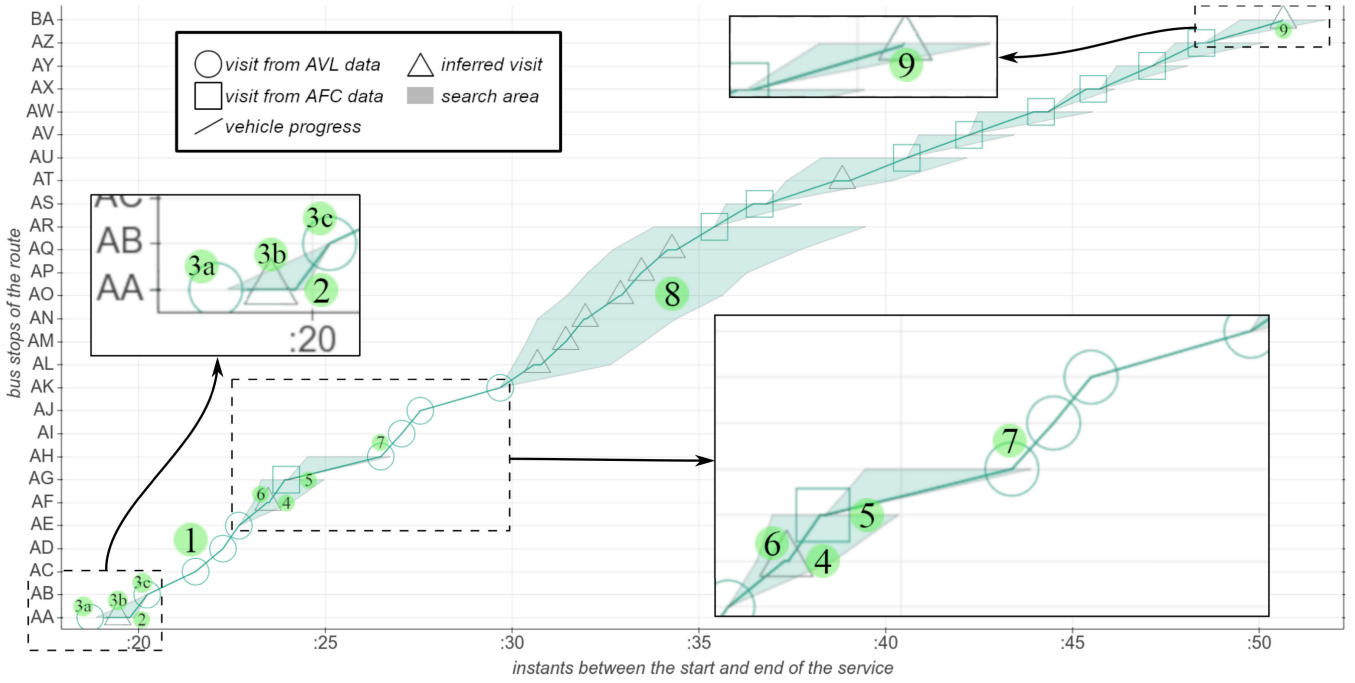



**FIGURE 5.** Trip inference process.

entry ($\tau_\phi$); and, correspondingly, on the lowest one from the "latter" ($\tau_\lambda$):

- If $\tau_\phi = \tau_\lambda = \tau$, they both represent the same stop, where the IPTS has records with both the old and the new *vehicle ids*. The following time ranges are computed at said stop:
  - A "feasibility range" $\zeta_{\phi,\lambda}$ that delimits the time span in which it is possible for the bus to have arrived after departing from the $(\tau - 1)^{\text{th}}$ stop, as described in the "former" trip $\phi$, and still make it to the $(\tau + 1)^{\text{th}}$ from the "latter" $\lambda$, considering the minimum bounds of the durations of the involved travel legs:

$$\zeta_{\phi,\lambda} \in R;\ \zeta_{\phi,\lambda} = \left(\upsilon_{\phi,(\tau-1)}[1] + \mu_{(\tau-1)},\ \upsilon_{\lambda,(\tau+1)}[0] - \mu_{(\tau)}\right) \quad (34)$$

  - A "bus presence range" $\eta_{\phi,\lambda}$, which is the minimum-span range that encompasses those of both the "former" and "latter" trips:

$$\eta_{\phi,\lambda} \in R;\ \eta_{\phi,\lambda} = \left(min\left(\upsilon_{\phi,\tau}[0], \upsilon_{\lambda,\tau}[0]\right),\ max\left(\upsilon_{\phi,\tau}[1], \upsilon_{\lambda,\tau}[1]\right)\right) \quad (35)$$

  The condition is met if these two ranges overlap:

$$\phi\ C\ \lambda\ \wedge\ \tau_\phi = \tau_\lambda \implies \zeta_{\phi,\lambda}\ \&\&\ \eta_{\phi,\lambda} \quad (36)$$

- If $\tau_\phi < \tau_\lambda$, the time span between the former trip's recorded departure from stop $\tau_\phi$ and the latter's registered arrival at $\tau_\lambda$ should be greater of equal than the lower bound of the total travel time between them.
- If $\tau_\phi > \tau_\lambda$, the two candidate trips do not originate from a single one that changed its id once.

Equation (37) summarizes these criteria:

$$\phi\ C\ \lambda \Leftrightarrow \begin{cases} \zeta_{\phi,\lambda}\ \&\&\ \eta_{\phi,\lambda} & \text{if } \tau_\phi = \tau_\lambda \\ \upsilon_{\lambda,\tau_\lambda}[0] - \upsilon_{\phi,\tau_\phi}[1] & \\ \quad \leq \sum_{\tau=\tau_\lambda}^{\tau_\phi - 1} \mu_i & \text{if } \tau_\phi < \tau_\lambda \end{cases} \quad (37)$$

### 2) UPDATE TRIPS CHARACTERIZATION TABLES

The trips that comply with eqs. (32), (33) and (37) are merged in a new one. Arrival and departure times at any stop between $\tau_\phi$ and $\tau_\lambda$ will be chosen as the most likely ones, according to dwell and travel time distributions; and the information to link them is stored in the columns *merged_trip* and *merged_trips*. Figure 6 illustrates an example, where the methodology will detect that entries that were on a first approach used to assert that two different trips of a route between stops AR and BJ took place ( blue and orange ) are actually part of the same one, and then re-evaluate unknown calls where this fact may be used to improve arrival and departure estimations:

1 The two trips proposed by the section III-E of this methodology comply with the conditions that identify them as a single one, with an intermediate *vehicle id* change:
- Their time buffers overlap in at least a stop, as can be seen observing the parts colored blue and orange .
- Considering only calls backed by IPTS entries, the latest from one of the trips (visit of blue at AY, 17:11:50, 1a) happens in a stop prior to the earliest from the other (visit of orange at BE,

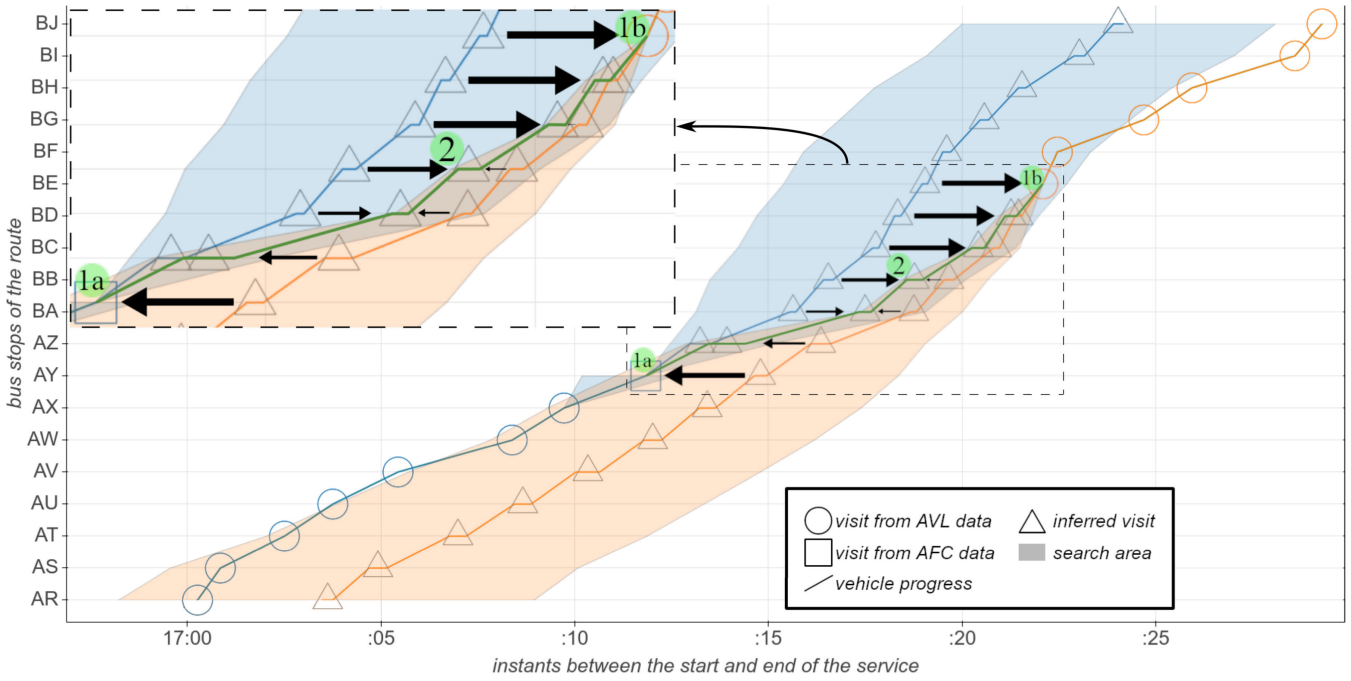



**FIGURE 6.** Merging of 2 trips (former blue, latter orange, modified visits green line).

17:22:05, 1b). The span between the departure from the former and the arrival at the latter is 10m15s, while the sum of the lower bounds of the trip legs involved is 2m23s, which means that a single vehicle could be responsible for both.

2 Intermediate arrival and departure times between AY and BE are re-calculated. Instead of their mean values according to their respective distributions and the departure from AY or the arrival at BE; they will adopt the most likely combination of values that satisfy both conditions at the same time.

### G. ASCRIBE TRIPS TO SCHEDULED RUNS AND UPDATE VISIT TIME SPANS

This part of the methodology has several goals: firstly, to differentiate between planned trips that were materialized or not; to identify non-scheduled, extra runs; and to remove inferred *visits to stops* that did not actually take place, for those trips that are successfully identified as starting downstream the initial terminus stop.

After a trip has been linked to its scheduled beginning, the additional information from the *schedule* table may be used to further refine arrival and departure times. These are the proposed steps, also shown as a flowchart in fig. 7:

- A loop is performed over all (*scheduled beginning*, *trip*) pairs where the latter's *departure* from the planned stop falls within the former's buffer $q$, considering those that share the same *vehicle id* first, and then ordered by the absolute value of the time span between the trip's departure and the scheduled start, ascending. Unless either of them has already been linked, they become so with each other.
- If a pairing was found, starting at the initial terminus of the whole route, inferred *visits to stops* are consecutively removed from the trip, until one that it backed by AFC or AVL records is reached.
- If the planning subsystem registered the start of the trip, the plausibility of its corresponding time range $n$ will be evaluated, utilizing the appropriate feasibility range stored in the *search_ranges* table (if not available, one is computed utilizing the closest downstream data-supported call of the trip). If $n$ is judged credible, two situations may occur:
  - If the initial call of the trip was previously deduced from other IPTS data, the available information will be combined to obtain the earlier and latest presence of the bus at that stop.
  - Otherwise, $n$ will be used as the [arrival, departure] range at the beginning of the trip.
- Downstream inferred visits, up to the first one sustained by IPTS data, are improved to their new most likely values, considering the total travel time between the scheduled trip start and that first known data point, and travel and dwell time distributions.

Figure 8 shows the first stops from an example trip:

1 Its initial estimation has been linked to a planned start at stop AF, with a gap between their inferred and planned departure times of 51s.

2 The stops upstream the planned start are not backed by any IPTS records, and are erased.

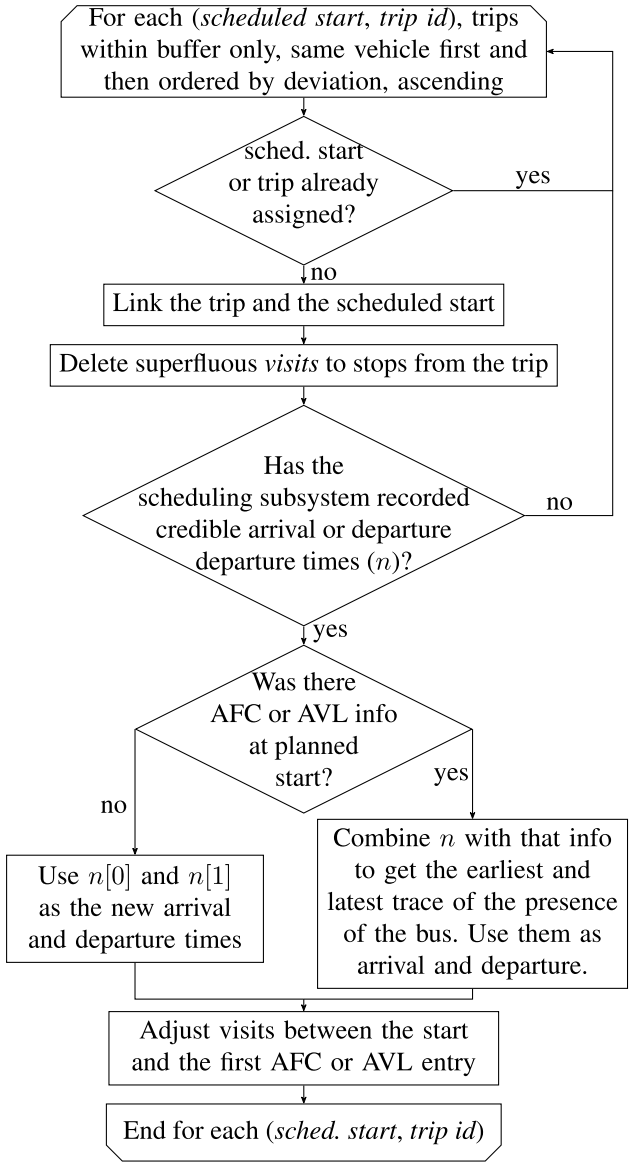



**FIGURE 7.** Associating scheduled and inferred trips.

3 In this case, the arrival and departure were logged by the scheduling subsystem at 07:25:39 and 07:26:01, respectively. Since these times falls within the *search range* for that trip at stop AF ([07:23:39, 07:26:13]), they are accepted as what really happened.

4 Visits to AG, AH, AI, and AJ are also recalculated, taking into account the new information.

### H. BOARDING GROUPS IMPUTATION

Once the calls of all possible trips have been defined and refined, boarding groups will be firstly mapped to a trip, and then to the stop where they took place.

For the first task, an imputation range (lightly marked for the latter case in fig. 9 with this pattern: ▨) is created for each trip from the moment when the vehicle arrived to its initial stop, minus the upper bound for the headway of

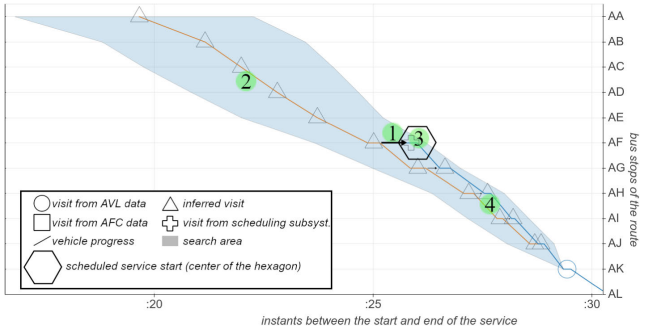



**FIGURE 8.** Improvement of arrival and departure times of a trip once its likely planned beginning is identified. Final characterization **blue**, removed or modified entries **orange**.

the route ($s$, to make sure that all pertinent AFC events at the initial terminus are identified, marked as 6a); to the moment it left its next-to-last stop (as no AFC events should be assigned to the last stop of a trip), plus the parameter $\boldsymbol{o}$, which allows for some **leeway between AVL and AFC events**, to cover cases such as validations after the vehicle leaves the stop or minor clock desynchronizations, identified as 6b.

$$\boldsymbol{o} : \text{AFC leeway} \quad \text{time} \tag{38}$$

Each *boarding group* is treated in a two-step process:

- Firstly, it is assigned to the trip whose time range it overlaps and that refers to the same *vehicle* and *route*. If no trip is found, the route id sameness requirement is dropped, to treat those cases where the ticketing subsystem state did not reflect the route the vehicle was really following. Any *boarding group* left will not be linked to a trip.
- Then, the proper stop within the trip is identified, considering all its calls but the last one:
  - If the gap between the *boarding_range* and the *visit_range* at the stop specified by the *boarding group* is less or equal to the maximum leeway $\boldsymbol{o}$, that stop will be accepted as the one where the travelers got on the bus (e.g., 3a in fig. 9).
  - Otherwise, it will be assumed that the AFC did not properly identify the *id* of the stop. The one from the closest call of the vehicle will be chosen instead (e.g., 3b in fig. 9, where the 3 *boarding groups* that were recorded as happening at stops BJ, BK, and BN are respectively assigned to BO, BP, and BS instead).

### I. SELECT TRIPS BACKED BY ENOUGH INFORMATION

The last step is to establish and apply criteria to accept or reject each of the possible trips that have been identified by this methodology. It is suggested to set boundaries that consider these features (eq. 39):

- Whether or not a planned departure was mapped to the trip ($w$). In the latter case, also consider if the *id* of the

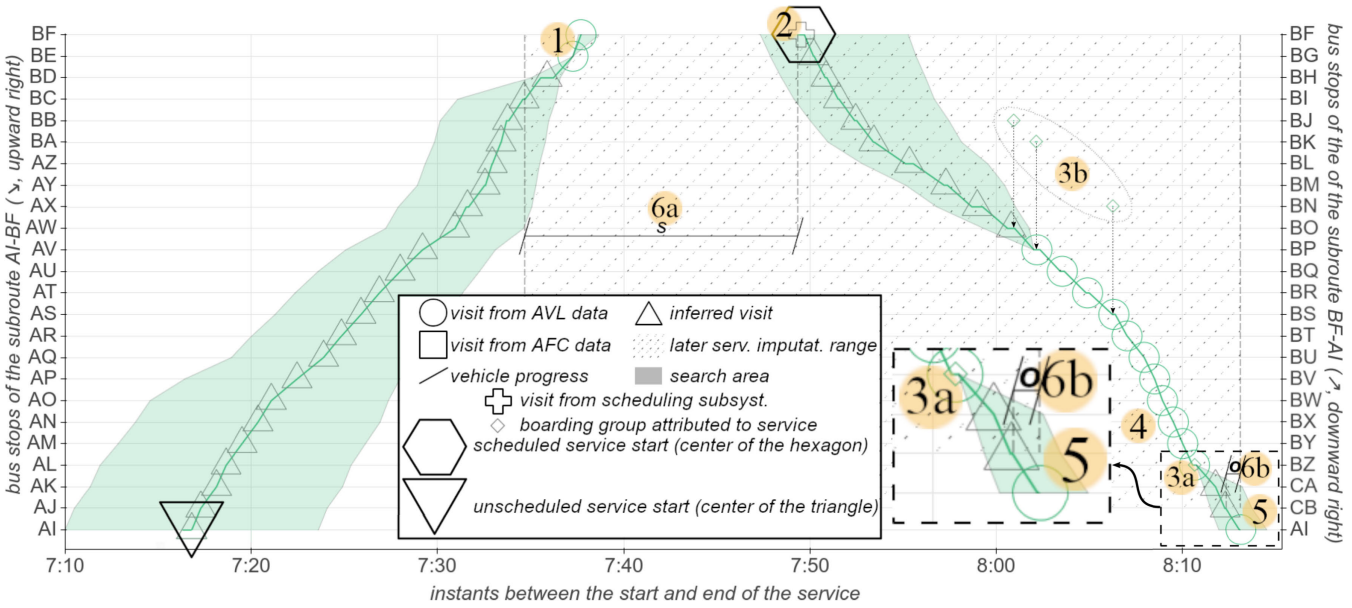



**FIGURE 9.** Imputation of boarding groups, and analysis of the IPTS data that sustain each of two consecutive potential trips of a vehicle.

vehicle is the same in both databases ($p$), and whether the scheduling subsystem registered a compatible stating time ($v$).

- The total number $h$ of *boarding groups* attributed to the trip, as described in section III-H.
- How many *visits* of that trip stem from AVL information ($f$).
- The number of stops between the earliest and latest visits supported by IPTS data ($l$).

$$\begin{aligned} &w : \text{trip was planned} && \text{boolean} \\ &p : \text{the planned vehicle was utilized} && \text{boolean} \\ &v : \text{the scheduling subsystem registered} \\ &\qquad \text{a valid starting time} && \text{boolean} \\ &h : \text{boarding groups count} && h \in \mathbb{N} \\ &f : \text{visits backed by AVL data} && f \in \mathbb{N} \\ &l : \text{longest range between stops} \\ &\qquad \text{backed by IPTS data} && l \in \mathbb{N} \end{aligned} \tag{39}$$

Figure 9 provides an example, analyzing two consecutive possible trips of a vehicle, covering complementary subroutes between AI and BF termini, both composed of 23 trip legs. Only 2 consecutive entries from the *avl_coalesced* table hint at the existence of the earlier (1); while the latter is supported by a planned trip of that vehicle for which the scheduling subsystem recorded the first call (2), 4 *boarding groups* (shown in 3a and 3b), 12 *avl_coalesced* rows (4), and by the fact that the span between its earliest (2) and latest (5) calls obtained from recorded IPTS observations covers the whole route. The former almost certainly did not happen, while the latter most likely did.

**FIGURE 10.** Santander city [56], [57].

## IV. CASE STUDY

The results of this methodology are illustrated utilizing the AVL and AFC events, and scheduled trip beginnings from the vehicles that, for 1 year, run route 1 in Santander, a city on the northern coast of Spain (fig. 10).

It operates from approximately 07:00 to 23:00, with headways of at most $s = 20$ min. In approximately half of the occasions, the scheduling subsystem records, with a deviation of around $z = 20$ s, the arrival and departure of the vehicle from the first stop of the trip. A complete round trip requires at least $d = 1$ h, while a single trip leg, even in the most unfavorable circumstances, should not take more than $e = 15$ min. While the IPTS is extremely helpful during day-to-day operations, the exploitation of its data must overcome several issues:

- Low AVL and AFC reliability at most trips' beginnings ($y$ = true), due to how on-board computers are sometimes operated and to the fact that when a bus is empty as it approaches the end of the route, drivers often


**FIGURE 11.** Bus stops of route 1.

find more convenient to wait until their next run in a stop upstream from the final one.

- Daily, each trip covering one of the 2 subroutes sometimes cannot be reliably identified with an id within the AVL and AFC datasets: this field may show several values within a single trip, or the same value may be used for consecutive runs covering both subroutes. Also, this id is not consistent between the AVL, AFC, and planning information.
- Missing AVL entries.
- Wrong AVL and AFC events that stem from the limitations of the IPTS, such as GPS signal loss, communication failures, or on-board computer errors; or from atypical or incorrect operations (e.g., setting vehicle state parameters that mistakenly identify the task being performed).
- The information regarding whether a planned trip finally happened and when did it start is most of the times accurate, but sometimes a normally performed trip fails to register, or it does with highly inaccurate timestamps.
- The available vehicle ids are associated with each driver-bus pair. Thus, those few daily occasions where workers end their shifts mid-trip will present 2 values.

This implementation utilizes the procedural language PL/pgSQL within a PostgresSQL 13.2 database for its core tasks; and Python 3.8 and Bokeh 2.2 to show an interactive representation of the results.

### A. IMPLEMENTATION OF THE METHODOLOGY

#### 1) INPUT DATA

##### a: BUS STOPS AND SUB-ROUTES

Santander has approximately 460 bus stops. The location of the 75 ones that shape route 1, which is divided in two sub-routes with one intermediate stop ('Consuelo Berges 16') and both termini in common, is shown in fig. 11. These sub-routes provide the templates which will be used to break down the *stop sequences* found during the treatment of the AVL data.

This itinerary begins at the Pctcan science park in the west, and traverses the city eastward through main arteries, passing by many of its commercial, residential, touristic, and administrative centers until it reaches La Península de La Magdalena Park (one of its foremost leisure locations). Then, it turns north-westward, and follows the coastline, providing access to Santander's most popular beaches. Finally it ends in Valdenoja, a neighborhood that even though presents some limited commercial use can be characterized as a dormitory suburb.

During non-business days the activity at Pctcan greatly diminishes, so buses do not visit the 3 easternmost stops. Also, especially during working days, several planned but not announced reinforcement trips begin downstream the first stop, to make use of short free slots drivers have between other assignments.

##### b: AFC

The dataset includes 2 586 600 raw AFC events. Almost all (99.99 %) correspond to real stops within the city, while the rest have *ids* that do not refer to a physical stop.

##### c: AVL

There are 1 569 417 raw AVL events. All represent calls at real stops of the city.

##### d: SCHEDULING INFORMATION

While the daily timetable that travelers consider when planning their trips on route 1 specifies, depending on whether it is a business day or not, around 100 or 80 places and times where a trip begins, the transport authority plans some extra actual vehicle runs, offering less-known additional trips of the route, such as several starting at Valdecilla hospital for staff that just ended their shifts, or reinforcing the offer during known peak demand periods when the distribution of available resources allows to do so. In approximately 95% of occasions a detected trip start time was logged. Extra vehicle runs not present in the scheduling information may occur due to tactical decisions during day-to-day operations.

#### 2) PREPROCESSING

##### a: AFC

Following the methodology outlined in section III-B1, 719 971 *stop groups* have been found. Using a value of 20 min for the parameter $s$, the maximum headway for this route, leads to splitting them in 724 550 *boarding groups* (0.6 % more events). Of these, 108 (0.01 %) last more than $s$ and will not be considered. There is, on average, 1 *boarding group* per 4 passenger boardings. Moreover, they provide a first fallback estimation of arrival and departure times at the stops, which will be utilized if no AVL records are available.

##### b: AVL

As explained in section III-B2c, consecutive AVL events that represent the same visit to a stop are merged, leaving 1 532 299 entries (2% less). Of these, 78 520 (5%) are deemed unreliable because they are part of impossibly short trip

legs. The remaining 1 453 779 entries, gathered in the table avl_coalesced, are classified in 45 840 trajectories.

#### 3) ANALYZE AVL TRAJECTORIES AS SEQUENCES

The 45 840 trajectories present 5800 different sequences of stops. The two most frequent ones match the already known itineraries of the subroutes under study (fig. 11), accounting for around 30% of the trajectories. Others contain in most cases one or several fragments compatible with one of the subroutes (as described in table 8d), though sometimes (2% of trajectories) the state of a vehicle did not change between subroutes, so a single trajectory contains information regarding more than one trip.

#### 4) SPECIFY TRAVEL TIMES AND DWELL TIMES DISTRIBUTION MODELS

Due to its computational advantages, two families of Normal distributions (eq. 40) have been chosen to model leg travel and dwell times. Considering the mobility cycles of the city, each of these families provides a different function for each subroute, stop, type of day (working, Saturdays, or Sundays and holidays), period of year (summer or not), and time bin (with a span of 30 min, and approximately 16 daily hours of trip, there are 32 possible time buckets: 07:00 to 07:30, 07:30 to 08:30, and so on).

$$
\begin{aligned}
p_{a,\tau,\gamma,\delta,\zeta,\eta} &: \text{trip leg travel time} \quad t \in T; \\
&\quad t \sim \mathcal{N}\left( \left(\mu_p\right)_{a,\tau,\gamma,\delta,\zeta}, \left(\left(\sigma_p\right)_{a,\tau,\gamma,\delta,\zeta}\right)^2 \right) \\
u_{a,\tau,\gamma,\delta,\zeta,\eta} &: \text{dwell time} \quad u \in T; \\
&\quad u \sim \mathcal{N}\left( \left(\mu_u\right)_{a,\tau,\gamma,\delta,\zeta}, \left(\left(\sigma_u\right)_{a,\tau,\gamma,\delta,\zeta}\right)^2 \right) \\
a &: \text{route id} \quad \text{From the methodology (➦)} \\
\tau &: \text{stop number} \quad \text{From the methodology (➦). For trip legs, their first stop.} \\
\gamma &: \text{period of year} \quad \gamma \in \{\text{'summer'}, \text{'rest of the year'}\} \\
\delta &: \text{type of day} \quad \delta \in \{\text{'working'}, \text{'Saturday'}, \text{'Sunday or holiday'}\} \\
\zeta &: \text{time of day bin} \quad \zeta \in \{1 \ldots \eta\} \\
\eta &: \text{time bins in a day} \quad \eta \in \mathbb{N}
\end{aligned} \tag{40}
$$

Route 1's leg travel times and dwell times have been characterized at each stop by roughly 2 periods $\cdot$ $3 \frac{\text{day types}}{\text{period}}$ $\cdot$ $32 \frac{\text{distributions}}{\text{day type}}$ $=$ 192 distributions each. Their means and standard deviations have been calculated utilizing the pertinent entries from table *avl_coalesced*.

#### 5) ASSEMBLE TRIPS

After applying the process described in section III-E, setting its parameters to $\boldsymbol{g} = 0.998$ and $\boldsymbol{c} = 2$ stops, 42 319 possible trips were found.

#### 6) MERGE INSTANCES WHERE A VEHICLE CHANGED ITS ID MID-TRIP

This refinement leads to the detection of around 2 daily occurrences of this issue, reducing the number of candidate trips to 41 641.

#### 7) ASCRIBE TRIPS TO SCHEDULED RUNS AND UPDATE VISIT TIME SPANS

40 352 trips have been mapped to a scheduled trip beginning (111 utilizing a vehicle different from the planned one); while the other 1289 were not. 86% of logged trip start times were utilized to characterize the first call of their trips.

#### 8) BOARDING GROUPS IMPUTATION

Applying the criteria described in section III-H, using an AFC leeway of $\boldsymbol{o} = 1$ min, provides the following results:

- 94.7 % of all boarding groups have been deemed to be correctly reporting their route and bus stop.
- 5% have been assigned to other stop than the automatically logged one.
- 0.3 % were not linked to a trip. They likely represent instances when the state of the vehicle incorrectly reported that it was traveling route 1.

#### 9) SELECT TRIPS BACKED BY ENOUGH INFORMATION

After considering the results from sections IV-A7 and IV-A8, the following acceptance criteria have been chosen (utilizing the nomenclature from eq. 39):

- For trips mapped to a scheduled beginning ($w =$ True):
  - Always accept if the planned vehicle was utilized ($p =$ True).
  - If a bus other than the scheduled one was used ($p =$ False), require at least 3 boarding groups linked to the trip ($h \geq 3$).
- Unscheduled trips will require stronger evidence: at least three ticketing events and no less than 12 total entries (one third of the number of stops of a subroute) endorsing its existence ($h \geq 3 \ \wedge \ h + f \geq 12$)

Applying these thresholds, the methodology reports on average 120 and 97 daily trips, depending on weather analyzing a business day or not. In the former case, the 96.5% of trips had previously been planned, and were materialized with the intended vehicle; while 3% were planned, but executed with a different vehicle; and 0.5% were unplanned trips. During non-business days, the corresponding ratios are 99.2%, 0.5%, and 0.3%; which are consistent with weekends and holidays being usually less demanding for the public transport of the city, resulting in less deviations from the schedule to react to the evolution of the traffic system.

Of all raw AVL data available, 92% was finally used to provide information to re-create a call of a trip. The source utilized to discern bus calls was AVL, statistical inference, a trip beginning logged by the scheduling subsystem, and AFC in 91%, 7%, 2%, and 1% of occasions.

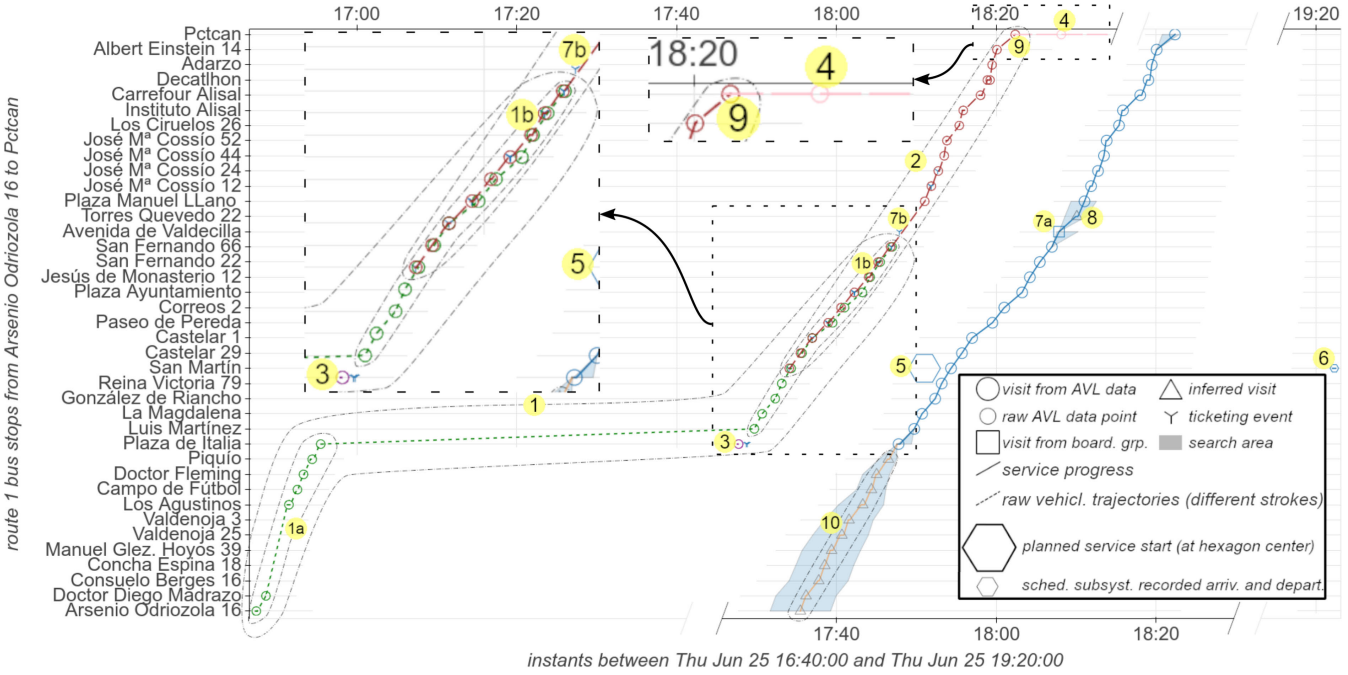



**FIGURE 12.** Characterization of a trip from fragmented and erroneous information.

## B. DISCUSSION

### 1) TRIP DEFINITION

This section gathers several examples to illustrate how this methodology has successfully improved the characterization of trips that were registered in the IPTS in a way that impeded their consideration.

#### *a: RECONSTRUCTION OF A TRIP FROM FRAGMENTED AND ERRONEOUS INFORMATION*

Figure 12 shows the case chosen for this analysis. The temporal horizontal axis has been broken in three regions with a shift between them for easier visualization:

- The central one, where the actual trip detected by the methodology and the planned departure (5) are depicted. Its temporal axis has been placed in the lower part of the plot.
- The leftmost area, with its temporal axis located in the upper part of the figure. It includes the relevant raw AVL and AFC data, with a −20 min shift:
  - 4 AVL *group UIDs*:
    - 1: From 'Arsenio Odriozola 16' to 'San Fernando 66,' with a gap of almost 1 h between 'Plaza de Italia' and 'Luis Martínez'.
    - 2: From 'San Martín' to 'Pctcan,' overlapping with 1 along its first 9 stops, and missing data at 'Avenida de Valdecilla' and 'Torres Quevedo 22.'
    - 3: A single event, at 'Plaza de Italia'.
    - 4: A single event, at 'Pctcan', the last stop of the trip. It happens around half a minute before 2 ends.
  - 19 AFC events, occurring between 'Plaza de Italia' and 'José Mª Cossío 24.'
- The rightmost zone only contains the clearly unrelated arrival and departure times logged by the planning subsystem (6), with a −40 min shift.

The proposed trip has been constructed making use of the available information. The first part of trip 1 was considered as 2 different fragments, discarding the earlier (1a, which was probably caused by an incorrect vehicle state) and utilizing the latter (1b). After the last entry from 1, the call at 'Avenida Valdecilla' (7a) is approximated from a ticketing event (7b); and the one at 'Torres Quevedo 22' (8) is inferred considering departure and arrival times from the previous and next stop, respectively. Of the two possible arrivals at the final terminus (9), the one from trip 4, which happens 30 s earlier, is more likely according to the departure time from 'Albert Einstein 14' and the travel time distribution between these stops during the time period [17:30-18:00] on a workday.

It is worth noticing that even though the trip was scheduled to start at 'San Martín,' the methodology has successfully detected that it actually began a few stops upstream (at 'Plaza de Italia,' from trip 3). The search for previous events (10) did not return any match, so that is the stop where the trip began.

#### *b: VEHICLE ID MID-TRIP CHANGE*

Figure 13 shows how the information regarding a trip of the subroute from Pctcan to Arsenio Odriozola appears in the IPTS, and its characterization by this methodology. Again, the horizontal temporal axis has been divided in three zones:

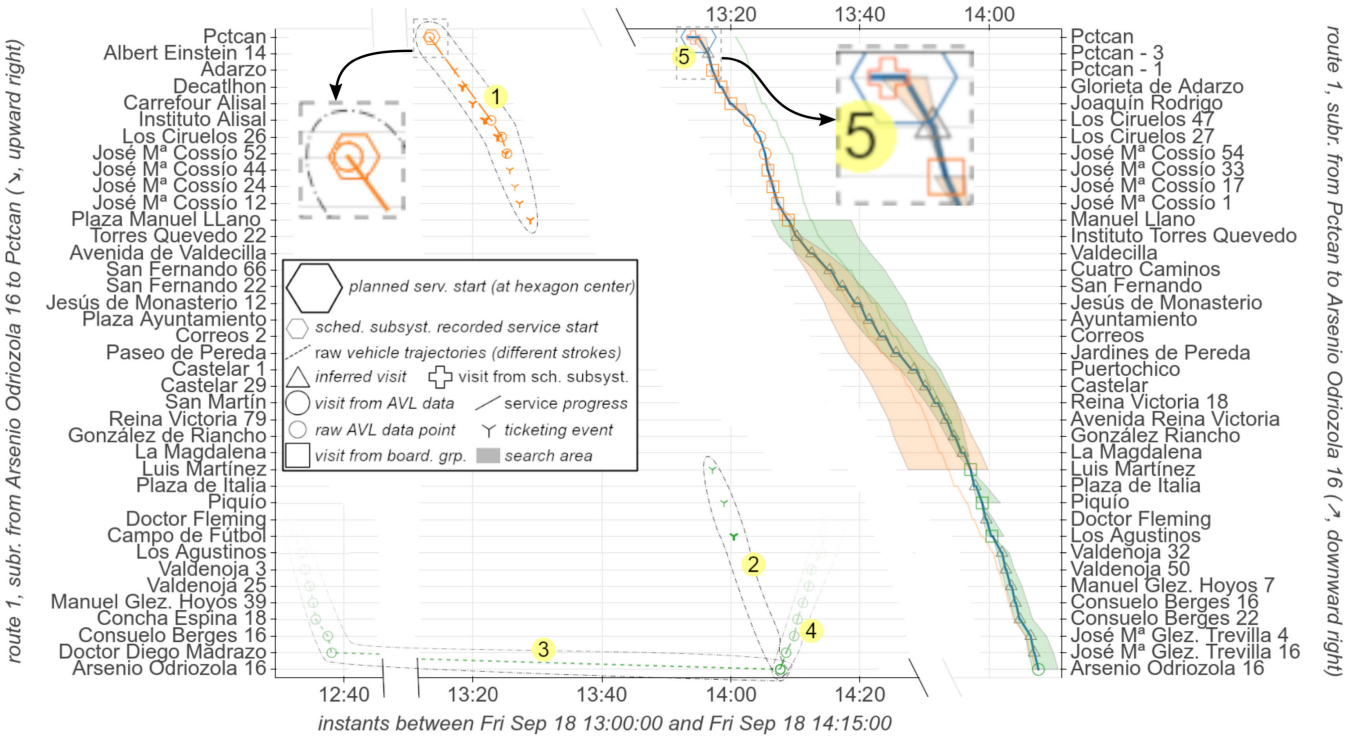



**FIGURE 13.** **Characterization of a trip when its vehicle id changes while it happens.**

- The rightmost area, which contains, with the temporal axis on top, the two trips initially detected, how they have been combined, and the planned start linked to them.
- The middle and leftmost regions show, with shifts of −40 min and −20 min and their temporal axes at the bottom, the pertinent raw records.

Initially, step III-E had found two trips:

- One for vehicle 14 ( orange ), backed by a 4-stops trajectory, and several ticketing events ( 1 ), being the latest one at 'Manuel Llano.'
- Another for vehicle 224 ( green ), inferred from 4 ticketing events at 3 stops ( 2 , the earliest at 'Luis Martínez'), and any of the two raw AVL events with the same timestamp at 'Arsenio Odriozola 16' terminus, which are part of opposite trajectories which end ( 3 ) or begin ( 4 ) there.

These have been detected, as described in section III-F, to be part of a single trip (displayed with a thicker blue line). Its corresponding scheduling subsystem entry ( 5 ) only detected the departure of the vehicle, a bit later than the available AVL data at that stop. Since it falls within the feasibility range from 'Pctcan - 1,' it is accepted and used to update the departure time at 'Pctcan,' and to improve the inferred call at the intermediate stop 'Pctcan - 3.'

### *c: NO AVL DATA AND WRONG VEHICLE ID*

Figure 14 shows a case that illustrates two situations that happen in the use case: the AVL subsystem not recording any entry, and a vehicle different from the planned one carrying out the trip.

There is a shift of 10 min between where the trip and the scheduled departure are drawn (rightmost part, temporal axis on top), and where the raw AFC data can be found (on the left, temporal axis at the bottom). It can be seen ( 1 ) that, since the scheduling subsystem did not register the beginning of the trip, the calls and 'Pctcan' and 'Pctcan - 3' had to be inferred using the arrival at 'Pctcan - 1' as the fix.

### 2) TREATMENT OF INITIAL TERMINI

The objective of this section is to study the benefit of the way this methodology handles the data available at particularly problematic termini, as it happens in this route. To this end, the 25 466 trips which present recorded starting times from the planning subsystem that, as described in section III-G, have been accepted for their characterization, will be used as the ground truth to be compared with the results obtained in three scenarios where that information will not be considered:

- A Follow the default methodology behavior for a route when the scheduling subsystem did not record the start of a trip (➦).
- B If the data at the first stop is deemed feasible, utilize it in the same way as any other stop.
- C if the planned start of a trip falls within its corresponding feasibility range (already stored in the *search_ranges* table, or computed utilizing the closest downstream data-supported call of the trip), it will be used as departure, if it happens later than any available AFC

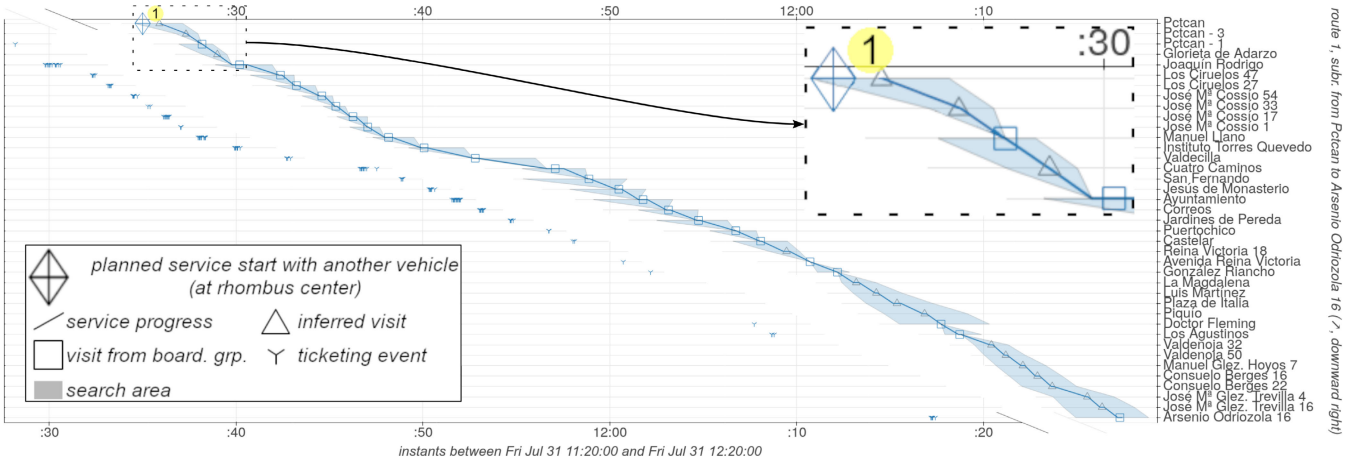



**FIGURE 14.** Trip characterization from AFC data only. Actual vehicle not the planned one.

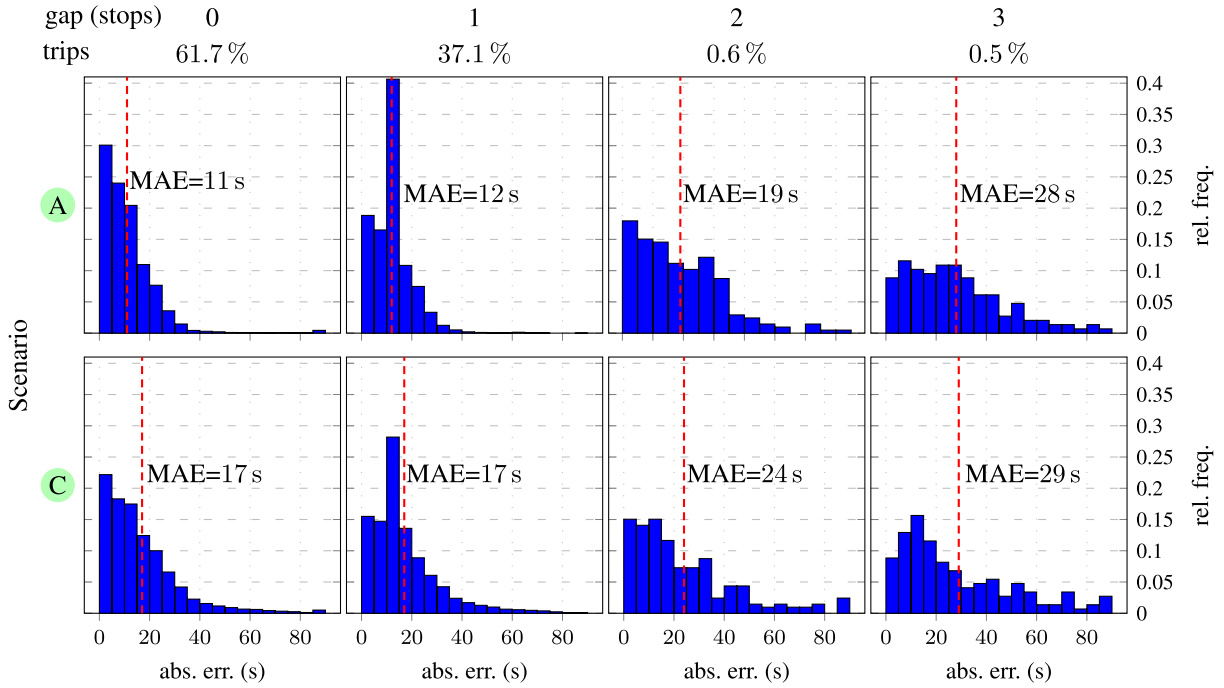



**FIGURE 15.** Distributions of abs. departure time errors per gap for scenarios A and C.

or AVL entries. This means assuming that schedule adherence is high enough to trust the planned departure times, unless they are impossible or very unlikely.

Figure 15 shows the distributions of the absolute error of the trip departure time reported in scenarios A and C. Trips have been classified according to their "gap": how far away (measured in trip legs) are their earlier visits based on AVL or AFC data from their scheduled beginnings. As can be seen, the decision of relying on the inferred start time rather than the planned one provides approximations with less dispersion (standard deviations of 13 s and 17 s, respectively) and a smaller mean absolute error (MAE), though as the uncertainty increases (more unknown calls between the start of the trip and the first data point) this benefit lessens.

Scenarios A and B only differ for those trips where compatible AVL or AFC data at the scheduled first stop can be found (zero gap). Figure 16 shows their distributions of absolute errors in this case. Again, scenario A infers the missing data with less dispersion (std. devs. of 15 s and 17 s, respectively) and MAE (11 s versus 13 s).

### 3) ROBUSTNESS AGAINST MISSING AND WRONG DATA

This section analyses how the methodology is affected by missing and erroneous AVL information and trip start detection (ticketing events are fully available in all scenarios). The 16 863 trips where all calls were fully recorded by the scheduling and AVL subsystems (49% of all) will be used as the ground truth; and compared with the results of

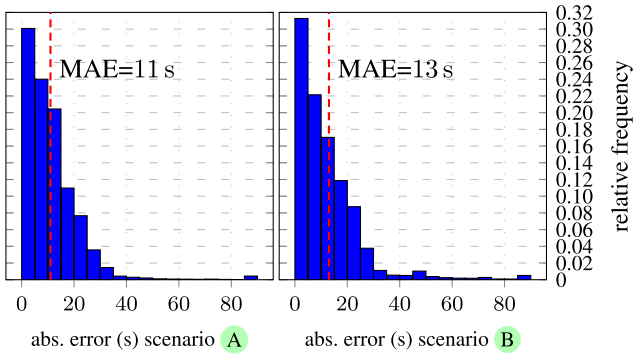



**FIGURE 16.** Distr. of abs. departure time errors when there is no gap for scenarios A and B.

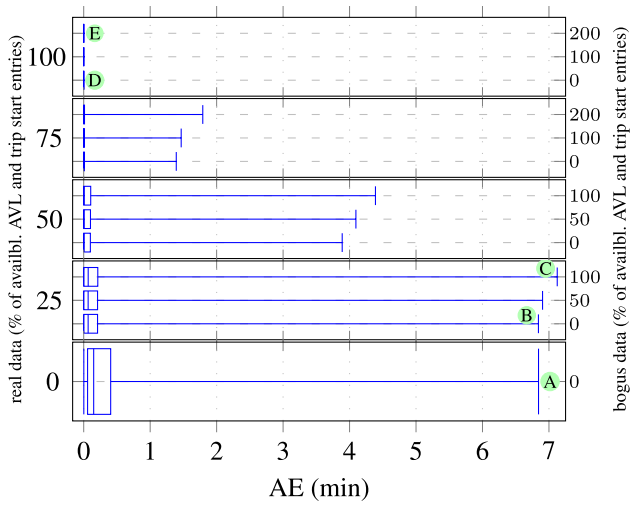



**FIGURE 17.** Distribution of the deviation from the ground truth for different proportions of real and wrong data. Extremes at 1st and 99th percentiles.

running this methodology utilizing only part of the recorded raw AVL data and scheduling subsystem detections, chosen through Bernoulli sampling; also adding different amounts of synthetic AVL erroneous readings, which have been randomly generated following these rules:

- *bus_stop*, *vehicle*, and *group* are chosen between all their distinct values.
- *instant* happens between 07:00 and 23:00 of any day of the year.
- Sampling from the distribution of *durations* is simulated utilizing its percentiles and the Uniform Distribution.

In fig. 17, the percentages are relative to the raw AVL entries and planned trips available in the dataset. For instance, a scenario with 25% of real data and 100% of simulated errors only reads the arrival and departure of the vehicles at the initial stop recorded by the scheduling subsystem in 25% of the scheduled trips, while its raw AVL input is created combining a Bernoulli sample of the real information with a probability 25% and 4 times as many bogus entries.

As more real data are available in a scenario, the more accurately trips are characterized. For instance, with a relatively small sample (25%), while the 99th percentile does not significantly differ from not using AVL or detected trip starts at all (slightly less than 7 min), it can already be appreciated that absolute error (AE) is quite more likely to be smaller: lower quartile, median, and upper quartile reduced from 4 s, 9 s, and 24 s to 0 s, 4 s, and 13 s, respectively (A & B).

It is also noticeable the strength of the methodology against artificial incorrect entries, which grows as more true readings are available in the scenario. Two examples are:

- With just 25% of real data, adding four times as many wrong entries only increases the 99th percentile from 6m51s to 7m07s (B & C).
- If all real information is available, the methodology successfully identifies the correct values as seeds, and is able to completely ignore many false events (D & E).

## V. CONCLUSION

The methodology described in this paper combines AFC, AVL, and scheduling subsystem information to provide a better characterization of the trips of the routes offered in a public transport system; ameliorating the problems that commonly occur when working with IPTS data: ambiguous ids for some elements of the system; missing or multiple entries related to the same AVL event; inconsistent trip ids between the different subsystems, which impedes identifying their respective records related to the same trip; AFC events with wrong information; and uncertainty regarding whether a programmed trip actually took place.

Events whose attributes wrongly classify them as part of different trips are identified and treated, as also are those unlikely to have really happened. Calls at each stop of each trip are delimited considering the multiple sources of data available in that particular instance, providing estimated arrival and departure times instead if there is none. A way to detect and handle those cases where the vehicle changes its id mid-trip, leading to the misrepresentation of their load profiles, is formulated. A trip and a stop are assigned to each ticketing record, distinguishing those cases where the AFC state information and timestamp are coherent with the corresponding trip call, and those where their timestamp and vehicle id will be utilized to infer the ticketing action that really took place. Finally, this methodology is applicable to situations with different scheduling information: none at all, planned beginnings only, or scheduled and detected (but not necessarily correct) start times; while the id of the originally intended vehicle may be known or not. These improvements help provide more accurate depictions of user rides and vehicle trips.

A case study has been presented, where several examples illustrate some of the issues this methodology solves: fragmented and erroneous information, vehicle id change mid-trip, and characterization of a trip utilizing AFC data only, where a vehicle different from the planned one was used.

To evaluate how it fares characterizing calls at the termini, those starting times recorded by the planning subsystem deemed to be correct have been used as empirical evidence; to compare with the outputs (without using that information) of the chosen strategy of preferring to infer the initial call in unreliable termini if the stop next to them is backed by real data, and two alternatives that were considered while writing this paper: treat termini as any other stop, and consider planned trip start times if feasible. It can be seen how the former consistently provides a better approximation of when trips have begun. This improvement may be particularly useful to better audit how closely the system adheres to its timetable.

Also, to assess the impact of bad AVL records, as well as missing AVL and detected trip beginning information, those trips perfectly recorded in the original dataset (start logged by the scheduling subsystem, and all other calls derived from AVL) will be used as the ground truth, studying how their characterization deviates with different amounts of real and bogus simulated entries. The results show significant improvements: with as little as 25% of real AVL and trip start detection data, even when adding 4 times as many wrong entries the results are significantly better than those from applying the methodology using only AFC records. The more real data is available, the closer the characterization is to the trip that took place, and the more resistant it is to incorrect values: for instance, if 100% of the real information is utilized, the methodology can completely filter out twice as many erroneous data.

As their next objective, the authors are currently working on the application of the trip chains methodology with the ticketing events and the trips characterized by this methodology, to provide more accurate vehicle load profiles and OD matrices. Other possible lines of investigation are the utilization of other distributions to model dwell and travel times, or the application of more detailed models to estimate arrival and departure from ticketing events when no AVL records are available.

## REFERENCES

[1] P. G. Furth, B. J. Hemily, T. H. J. Muller, and J. G. Strathman, "TCRP web document 23 (project H-28): Contractor's final report uses of archived AVL-APC data to improve transit performance and management: Review and potential," *TCRP Web Document*, vol. 23, pp. 8–12 and 21–22, Jun. 2003.

[2] M. Trépanier, N. Tranchant, and R. Chapleau, "Individual trip destination estimation in a transit smart card automated fare collection system," *J. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 1–14, 2007.

[3] M. Trépanier, C. Morency, and B. Agard, "Calculation of transit performance measures using smartcard data," *J. Public Transp.*, vol. 12, no. 1, pp. 79–96, 2009.

[4] N. H. Wilson, J. Zhao, and A. Rahbee, "The potential impact of automated data collection systems on urban public transport planning," *Oper. Res. Comput. Sci. Interfaces*, vol. 46, no. 1, pp. 75–99, 2009.

[5] X. Ma and Y. Wang, "Development of a data-driven platform for transit performance measures using smart card and GPS data," *J. Transp. Eng.*, vol. 140, no. 12, pp. 1–13, 2014.

[6] R. L. Bertini and A. El-Geneidy, "Generating transit performance measures with archived data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1841, no. 1, pp. 109–119, Jan. 2003.

[7] Y. Lin, X. Yang, N. Zou, and L. Jia, "Real-time bus arrival time prediction: Case study for Jinan, China," *J. Transp. Eng.*, vol. 139, no. 11, pp. 1133–1140, 2013.

[8] Google. *GTFS Static Overview | Static Transit | Google Developers*. Accessed: 2020. [Online]. Available: https://developers.google.com/transit/gtfs

[9] MobilityData. *General Transit Feed Specification*. Accessed: 2020. [Online]. Available: https://gtfs.org/changes/

[10] *OVapi*. Accessed: 2020. [Online]. Available: http://ovapi.nl/

[11] Mapzen Foundation and Interline Technologies LLC. Welcome • Transitland. *Transitland*. Accessed: 2020. [Online]. Available: https://www.transit.land/

[12] MobilityData IO. *OpenMobilityData—Public Transit Feeds From Around the World*. Accessed: 2020. [Online]. Available: https://transitfeeds.com/

[13] I. Gokasar and Y. Cetinel, "A New strategy for the diagnosis of the bus headways using AVL data," in *Proc. 6th Int. Conf. Models Technol. Intell. Transp. Syst.*, Jun. 2019, pp. 1–7.

[14] N. Hounsell and B. Shrestha, "A new approach for co-operative bus priority at traffic signals," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 6–14, Nov. 2012.

[15] H. Saghaei, "Design and implementation of a fleet management system using novel GPS/GLONASS tracker and web-based software-automatic vehicle locator (AVL), fleet management system (FMS), global positioning system (GPS), global navigation satellite system (GLONASS), Ge," in *Proc. 1st Int. Conf. New Res. Achievements Elect. Comput. Eng.* Tehran, Iran: Amirkabir Univ. Technol., 2016.

[16] F. Van Diggelen and P. Enge, "The world's first GPS MOOC and worldwide laboratory using smartphones," in *Proc. 28th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Tampa, FL, USA, 2015, pp. 361–369.

[17] D. Rančić, B. Predić, and V. Mihajlović, "Online and post-processing of AVL data in public bus transportation system," *WSEAS Trans. Inf. Sci. Appl.*, vol. 5, no. 3, pp. 229–236, 2008.

[18] M.-P. Pelletier, M. Trépanier, and C. Morency, "Smart card data use in public transit: A literature review," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 557–568, 2011.

[19] L. M. Kieu, A. Bhaskar, and E. Chung, "Passenger segmentation using smart card data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1537–1548, Jun. 2015.

[20] T. Kusakabe and Y. Asakura, "Behavioural data mining of transit smart card data: A data fusion approach," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 179–191, Sep. 2014, doi: 10.1016/j.trc.2014.05.012.

[21] B. Chidlovskii, "Mining smart card data for travellers' mini activities," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3676–3685, Nov. 2018.

[22] L. M. Kieu, A. Bhaskar, M. Cools, and E. Chung, "An investigation of timed transfer coordination using event-based multi agent simulation," *Transp. Res. C, Emerg. Technol.*, vol. 81, pp. 363–378, Aug. 2017, doi: 10.1016/j.trc.2017.02.018.

[23] K. K. A. Chu and R. Chapleau, "Enriching archived smart card transaction data for transit demand modeling," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2063, no. 1, pp. 63–72, Jan. 2008.

[24] F. Kurauchi and J. D. Schmöcker, *Public Transport Planning With Smart Card Data*. Boca Raton, FL, USA: CRC Press, 2017.

[25] D. Luo, O. Cats, and H. van Lint, "Constructing transit origin–destination matrices with spatial clustering," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2652, no. 1, pp. 39–49, Jan. 2017.

[26] J. B. Gordon, H. N. Koutsopoulos, and N. H. M. Wilson, "Estimation of population origin–interchange–destination flows on multimodal transit networks," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 350–365, May 2018, doi: 10.1016/j.trc.2018.03.007.

[27] J. Zhao, M. Frumin, N. Wilson, and Z. Zhao, "Unified estimator for excess journey time under heterogeneous passenger incidence behavior using smartcard data," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 70–88, Sep. 2013, doi: 10.1016/j.trc.2013.05.009.

[28] P. G. Furth, B. Hemily, T. H. J. Muller, J. G. Strathman, *Using Archived AVL-APC Data to Improve Transit Performance and Management*. Washington, DC, USA: The National Academies Press, 2006. [Online]. Available: https://www.nap.edu/catalog/13907/using-archived-avl-apc-data-to-improve-transit-performance-and-management

[29] Z. Chen and W. Fan, "Extracting bus transit boarding and alighting information using smart card transaction data," *J. Public Transp.*, vol. 22, no. 1, pp. 1–23, Jan. 2020.

[30] A. A. M. Alsger and B. Eng, "Estimation of transit origin destination matrices using smart card fare data school of civil engineering," Univ. Queensland, Brisbane, QLD, Australia, Tech. Rep., 2016.

[31] S. Robinson, B. Narayanan, N. Toh, and F. Pereira, "Methods for pre-processing smartcard data to improve data quality," *Transp. Res. C, Emerg. Technol.*, vol. 49, pp. 43–58, Dec. 2014, doi: 10.1016/j.trc.2014.10.006.
[32] E. Hussain, A. Bhaskar, and E. Chung, "Transit OD matrix estimation using smartcard data: Recent developments and future research challenges," *Transp. Res. C, Emerg. Technol.*, vol. 125, Apr. 2021, Art. no. 103044, doi: 10.1016/j.trc.2021.103044.
[33] G. Chen, X. Yang, J. An, and D. Zhang, "Bus-arrival-time prediction models: Link-based and section-based," *J. Transp. Eng.*, vol. 138, no. 1, pp. 60–66, Jan. 2012.
[34] Z. Dai, X. Ma, and X. Chen, "Bus travel time modelling using GPS probe and smart card data: A probabilistic approach considering link travel time and station dwell time," *J. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 175–190, Mar. 2019, doi: 10.1080/15472450.2018.1470932.
[35] H. Levinson, "Analyzing transit travel time performance," *Transp. Res. Rec.*, vol. 915, p. 1, Dec. 1983.
[36] X. Qu, E. Oh, J. Weng, and S. Jin, "Bus travel time reliability analysis: A case study," *Proc. Inst. Civil Eng. Transp.*, vol. 167, no. 3, pp. 178–184, Jun. 2014.
[37] M. A. P. Taylor, "Modelling travel time reliability with the burr distribution," *Procedia Social Behav. Sci.*, vol. 54, pp. 75–83, Oct. 2012.
[38] A. Chepuri, J. Ramakrishnan, S. Arkatkar, G. Joshi, and S. S. Pulugurtha, "Examining travel time reliability-based performance indicators for bus routes using GPS-based bus trajectory data in India," *J. Transp. Eng., A, Syst.*, vol. 144, no. 5, May 2018, Art. no. 04018012.
[39] M. M. Harsha, R. H. Mulangi, and H. D. D. Kumar, "Analysis of bus travel time variability using automatic vehicle location data," *Transp. Res. Procedia*, vol. 48, pp. 3283–3298, Jan. 2020, doi: 10.1016/j.trpro.2020.08.123.
[40] M. Li, X. Zhou, and N. M. Rouphail, "Quantifying travel time variability at a single bottleneck based on stochastic capacity and demand distributions," *J. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 79–93, Mar. 2017.
[41] K. K. Srinivasan, A. A. Prakash, and R. Seshadri, "Finding most reliable paths on networks with correlated and shifted log–normal travel times," *Transp. Res. B, Methodol.*, vol. 66, pp. 110–128, Aug. 2014, doi: 10.1016/j.trb.2013.10.011.
[42] B. Y. Chen, C. Shi, J. Zhang, W. H. Lam, Q. Li, and S. Xiang, "Most reliable path-finding algorithm for maximizing on-time arrival probability," *Transportmetrica B*, vol. 5, no. 3, pp. 253–269, 2017.
[43] R. Rajbhandari, S. I. Chien, and J. R. Daniel, "Estimation of bus dwell times with automatic passenger counter information," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1841, no. 1, pp. 120–127, Jan. 2003.
[44] H. Zhang, H. Cui, and B. Shi, "A data-driven analysis for operational vehicle performance of public transport network," *IEEE Access*, vol. 7, pp. 96404–96413, 2019.
[45] *Public Transport Lines*, Aimsun Next 8.4 Users' Manual, Aimsun, Barcelona, Spain, 2020.
[46] *1.17.6.4 Using Time Distributions*, PTV Vissim 2021 User Manual, PTV Group, 2020, Karlsruhe, Germany, p. 240.
[47] R. Z. Koshy and V. T. Arasan, "Influence of bus stops on flow characteristics of mixed traffic," *J. Transp. Eng.*, vol. 131, no. 8, pp. 640–643, 2005.
[48] S. Rashidi and P. Ranjitkar, "Approximation and short-term prediction of bus dwell time using AVL data," *J. Eastern Asia Soc. Transp. Stud.*, vol. 10, no. 10, pp. 1281–1291, 2013.
[49] X. Jiang and X. Yang, "Regression-based models for bus dwell time," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2858–2863.
[50] D. Luo, L. Bonnetain, O. Cats, and H. van Lint, "Constructing spatiotemporal load profiles of transit vehicles with multiple data sources," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 8, pp. 175–186, Dec. 2018.
[51] L. Moreira-matias, "Towards an AVL-based demand estimation model," Transp. Res. Board, USA, Tech. Rep. 2544, 2016, pp. 141–149.
[52] K. Buneman, "Automated and passenger-based transit performance measures," *Transp. Res. Rec.*, vol. 992, no. 992, pp. 23–28, 1984.
[53] A. Gschwender, M. Munizaga, and C. Simonetti, "Using smart card and GPS data for policy and planning: The case of transantiago," *Res. Transp. Econ.*, vol. 59, pp. 242–249, Nov. 2016.
[54] K. K. A. Chu, R. Chapleau, and M. Trépanier, "Driver-assisted bus interview: Passive transit travel survey with smart card automatic fare collection system and applications," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2105, no. 1, pp. 1–10, Jan. 2009.
[55] Z. Huang, L. Xu, Y. Lin, P. Wu, and B. Feng, "Citywide metro-to-bus transfer behavior identification based on combined data from smart cards and GPS," *Appl. Sci.*, vol. 9, no. 17, p. 3597, Sep. 2019.
[56] *File: Spain Map Modern.png—Wikimedia Commons*. Accessed: 2020. [Online]. Available: https://commons.wikimedia.org/wiki/File:Spain_map_modern.png
[57] *Santander: A Digital Smart City Prototype in Spain—SPIEGEL ONLINE*. Accessed: 2020. [Online]. Available: https://www.spiegel.de/international/world/santander-a-digital-smart-city-prototype-in-spain-a-888480.html

**JUAN BENAVENTE** is currently pursuing the Ph.D. degree in civil engineering with the University of Cantabria.

He has participated in several European projects, being his most prominent work on smart cities (real-time estimation of traffic conditions), and on the use of heterogeneous and innovative sources of information to improve mobility. Within his academic activity, he has collaborated in papers on optimization of regional school transport, and the impact of freight transport in urban areas, in addition to two-book chapters (LUTI, cycling loan points layout). His main research interests include the exploitation of intelligent public transport systems data, and optimization applied to mobility, such as bike loan points layout, regional school buses scheduling, and freight transport impact in urban areas.

**BORJA ALONSO** received the M.S. degree in civil engineering from the University of Cantabria, Spain, in 2007, and the Ph.D. degree in transportation engineering, in 2010.

He is currently an Associate Professor of transport planning with the University of Cantabria. He has published more than 20 articles in international indexed journals (JCR), presented various conference papers at national and international congresses, and co-edited and coauthored two complete books and several book chapters. His research interests include operational analysis of traffic networks, design of public transport systems, and models of user behavior.

**ANDRÉS RODRÍGUEZ** received the master's degree in mobile application development. He develops his work as a Ph.D. Student in the field of user behavior in urban car parks and models of choice of parking typology and the analysis of large volumes of data applying machine learning techniques.

He has published five articles in international indexed journals and has carried out multiple roles in several European projects regarding railway sector long-term planning and applying new ways to fuse heterogeneous data to promote sustainable mobility. He has attended numerous conferences and courses in the field of development in Java, Python, and web languages.

**JOSÉ LUIS MOURA** received the B.S. and M.S. degrees in civil engineering from the University of Cantabria, Spain, in 2001, and the Ph.D. degree from the Department of Transportation Engineering, University of Cantabria, in 2005.

Since 2014, he has been the Director of the School of Civil Engineering (E.T.S. Ingenieros de Caminos, Canales y Puertos), University of Cantabria. He is currently a Professor with the Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos, University of Cantabria. He has carried out an intense teaching and research activity with relevant results that are reflected in the number and importance of impact international publications, research projects, the transfer of results, and the occupied management positions. His most important research interests include transport systems optimization, transport and environment, design and management of transport networks, cycling mobility, and freight urban distribution and mobility in smart cities.