An Online Learning Approach to Eliminate Bus Bunching in Real-time

Morriea-Matias , Luis; Cats, Oded; Gama, Joao; Mendes-Moreira, Joao; Freire de Sousa, Jorge

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

An Online Learning Approach to Eliminate Bus Bunching in Real-Time

Luís Moreira-Matias[a,*], Oded Cats[c], João Gama[b,e], João
Mendes-Moreira[b,d], Jorge Freire de Sousa[f,g]

[a]*NEC Laboratories Europe, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany*
[b]*LIAAD-INESC TEC, 4200-465 Porto, Portugal*
[c]*Dep. Transport and Planning, Delft University of Technology, 2600 GA Delft, The Netherlands*
[d]*Dep. de Eng. Informática, Fac. de Engenharia, U. Porto, 4200-465 Porto, Portugal*
[e]*Faculdade de Economia, U.Porto, 4200-465 Porto, Portugal*
[f]*UGEI-INESC TEC, 4200-465 Porto, Portugal*
[g]*DGEI, Fac. de Engenharia, U. Porto, 4200-465 Porto, Portugal*

Abstract

Recent advances in telecommunications created new opportunities for monitoring public transport operations in real-time. This paper presents an Automatic Control framework to mitigate the Bus Bunching phenomenon in real-time. The framework depicts a powerful combination of distinct Machine Learning principles and methods to extract valuable information from raw location-based data. State-of-the-art tools and methodologies such as Regression Analysis, Probabilistic Reasoning and Perceptron's learning with Stochastic Gradient Descent constitute building blocks of this predictive methodology. The prediction's output is then used to select and deploy a corrective action to automatically prevent bus bunching. The performance of the proposed method is evaluated using data collected from 18 bus routes in Porto, Portugal over a period of one year. Simulation results demonstrate that the proposed method can potentially reduce bunching by 68% and decrease average passenger waiting times by 4.5%, without prolonging in-vehicle times. The proposed system could be embedded in a decision support system to improve control room operations.

*Corresponding author.
    Email address: luis.moreira.matias[at]gmail.com, +49 6221 4342261 (Luís Moreira-Matias)

1. Introduction

Major urban areas worldwide are intensely equipped with sensors able to monitor daily human activities. Vehicles, Smartphones, Inductive Loop Counter and License Plate Recognition are some examples of equipments that can collect this type of information in real-time . The recent development of communicational technologies such as GSM-based (Global System for Mobile Communications) 3G, Global Positioning System (GPS) and WiFi enable to **access** this data in *near* **real-time**.

These location-based technologies provide an unprecedented opportunity to develop large scale monitoring frameworks. Many industries are increasingly taking advantage of such data to improve their services and monitor their operations. Public Transportation (PT) companies offer a good showcase of this trend: they operate in a highly competitive environment where the analysis of real-time data could potentially lead to service improvements and ultimately increase their market share. Service **reliability** is key in maintaining their profitability.

In the last decades, Bus Dispatching Systems have been deployed in many PT companies worldwide. These systems were first used to monitor fleet operations. Automatic Vehicle Location (AVL) data collected by these systems was later used for offline evaluation of service performance and reliability. These systems became therefore indispensable for PT service providers. The inherent uncertainty associated with PT operations could be mitigated in real-time by the **Control Centre**. However, it is often labor-intensive as traffic dispatchers need to take numerous decisions in response to rapidly evolving conditions.

Many researchers highlighted the potential of the stored AVL data to provide insights on how to improve PT reliability, e.g. [67, 27]. In particular, previous work tended to focus on offline ex-ante evaluation. However, the real time availability of AVL data opens new research directions for improving PT reliability, namely introducing real-time decision models to support **Operational Control**.

PT reliability could be defined either in terms of punctuality – the extent to which operations adhere to the planned schedule – or in terms of regularity – the extent to which vehicles are evenly spaced, implying even headways, the

time interval between successive vehicles running on the same route [69]. In the case of high-frequency routes (headways of 10 minutes or less), regularity is the main indicator of service reliability since it is the main determinant of passenger waiting times [10]. Headways are inherently instable due to a positive feedback loop between the headway, the number of passengers waiting at the stop, dwell times and successive headways [20]. For example, a small delay provokes an increase in the number of passengers in the next stop. This in turn leads to an increase in the dwell time (bus service time at a stop) and consequently, it further increases the bus delay. At the same time, the next bus will have fewer passengers, shorter dwell times and will gradually catch up with the preceding bus. This snowball effect will result in pairs of buses forming a platoon as illustrated in Fig. 1. This phenomenon is denominated **Bus Bunching(BB)** [20, 43, 45].

The prevalence of BB is one of the most visible blueprints of an unreliable service. Two (or more) buses running together on the same route is an undeniable sign that something is going terribly wrong with the company's service. Operational Control can potentially address BB in real-time. This paper focuses on using both historical and real-time AVL data to deploy automatic control strategies, to mitigate BB while reducing the human workload required to make these decisions. It provides a **complete bottom-up methodology**, from fundamental theoretical aspects that capture the BB
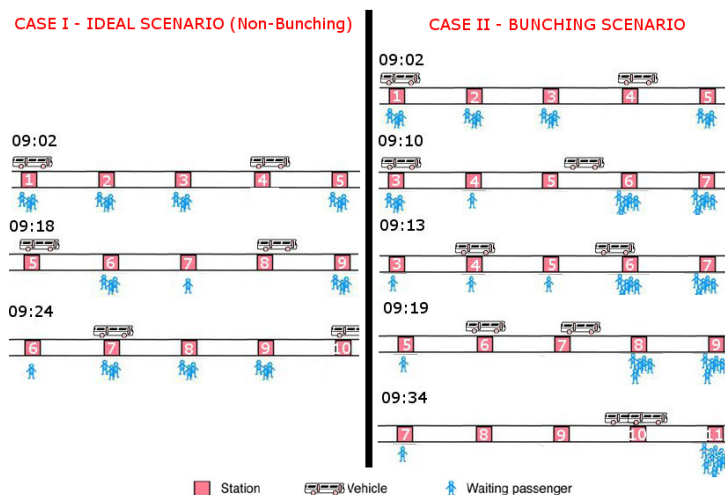

Figure 1: Bus Bunching illustration.

3

formation process to practical issues associated with the deployment of corrective actions, as well as with the evaluation of their impacts.

1.1. Literature Review

Previous studies have deployed a range of analytical and simulation models to represent the dynamics of the bus service operations and evaluate the impacts of alternative control strategies. Early analytical studies that examined the BB phenomenon and the characteristics of its instability triggered by recurrent perturbations include Newell and Potts [50], Chapman and Michel [17], Powell and Sheffi [55]. The latter devised a probabilistic model which builds a set of recursive relationships to calculate the p.d.f. to validate the hypothesis of generating a vehicle platoon formation at each stop. More recently, Daganzo [20] and Daganzo and Pilachowski [21] developed analytical models to assess the impacts of an adaptive control strategy which adjusts bus dwell times at stops and the running times between successive stops based on the respective headways.

Transit control strategies consist of a wide variety of operational methods aimed to improve transit performance and level of service. Holding strategies are among the most widely used transit control methods aimed to improve service regularity [2]. In order to implement corrective strategies and consequent actions, both the location - where the control decisions should be deployed [71, 1, 26, 68, 15] and the how – the criteria for intervening and its specification [32, 39, 13] - must be determined. In [24], a global control unit optimizes the holding times by solving a deterministic rolling horizon mathematical programming model which minimizes total passenger waiting times.

While previous studies might be effective in mitigating BB occurrences and therefore reducing service uncertainty, none of the abovementioned studies involved a systematic proactive approach to eliminate BB in real-time. This paper proposes online learning techniques to address headway instability by simultaneously considering historical and real-time data concerning service perturbations. Moreover, a comprehensive procedure for BB event detection and corrective action deployment is developed and applied.

1.2. Scope and Objectives

Most of the abovementioned state-of-the-art research on this topic departs from the assumption that the probability of BB events is minimized by

4

maximizing headway stability. This is achieved by either minimizing the difference between the actual headway and the scheduled one or by minimizing the discrepancies between successive headways. Notwithstanding its validity, this approach requires multiple control actions (i.e. speed modification, bus holding, etc.) which may impose high mental workload for both drivers and control center staff, yielding results with sub-optimal decision making and low compliance rates.

Hereby, we propose a **proactive** rather than a **reactive** operational control framework. The fundamental idea is to estimate the likelihood of a BB event occurring further downstream and then deploy a corrective control strategy.

The AVL data constitute an *unbounded* stream of data that arrives at a high rate. In the last decades, Machine Learning (ML) research has been essentially focused on *batch learning* using usually relatively small datasets. In *batch learning*, the training data is **assumed** to be entirely available for the learning algorithm. It outputs a decision model after processing the data once or multiple times. However, this particular problem requires a predictive methodology that can output values *while* the data is being collected (i.e. **incremental**), and to react to unexpected situations such as traffic jams or high demand events (i.e. **concept drift**) by adapting their learning models to the current state of the system.

This paper presents a modelling framework to prevent BB from emerging in real-time. It constitutes a powerful combination of distinct ML principles and methods to extract valuable information from a massive source of continuous data (such as the AVL). State-of-the-art tools and methodologies such as Regression Analysis, Probabilistic Reasoning [49] and Perceptron's learning with Stochastic Gradient Descent [60] are building blocks of this predictive methodology. The objectives of this paper are:

1. To develop a regression algorithm that is capable of producing **accurate predictions under concept drift**(s);

2. To elaborate a **probabilistic prediction** of downstream BB occurrences;

3. To select a **corrective action** to prevent BB occurrences and specify its **implementation**;

4. To evaluate the **impact** of the operational control on service reliability.

1.3. Main Contributions

The first three steps of this methodology were already briefly introduced in [45]. However, this work did not address the **Control** problem applied for mitigating these events. It served mainly as a proof of concept for modelling BB as an online supervised learning problem. Steps (4) and (5) of this methodology take full advantage on the interpretability given by the BB likelihood produced in step (I-3) of this framework, not only by recommending a corrective action, but also defining how these actions can be applied to real-world scenarios. Without such novel steps, the previous methodology would merely alarm on BB likelihood for a specific pair of buses. Consequently, the major contributions of this paper are:

1. a novel method for selecting the **most adequate corrective** action to be implemented in real-time based on the local headway probability distributions;

2. a formula to define the **actions deployment** using context-aware information - such as the BB forecasting horizon - and the BB likelihoods estimated for each stop;

3. a **parameter tuning** method to adequately fit the large set of parameters involved in this methodology (Section 5.2);

4. the framework is exhaustively validated using a **large-scale dataset** which includes vehicle positioning records collected over the course of a year for **eighteen routes** .

1.4. Paper Structure

This paper is structured as follows: Section 2 introduces some fundamental concepts in Public Transport Operations. The Data Collection employed in the experiments is described in Section 3, along with some details about its preprocessing, data structure and Case Study application. The fourth Section details the proposed stepwise methodology. Section 5 presents the experimental setup, introduces the evaluation metrics and a tuning framework to adjust the values of the methodology's parameters, as well as the artificial demand model employed to produce synthetic data about the passenger demand and the experimental results. Section 6 reflects on these

results and the potential implications of this framework for a real world Control center. Finally, some final remarks are presented along with suggestions for future research directions.

The symbols and notations used throughout this paper are provided in Tables 1 and 2.

2. Fundamental Concepts

This section introduces some fundamental concepts in PT Operations and Control. In particular, we present key evaluation metrics of PT reliability, discuss Travel Time Prediction (TTP) in the context of PT Planning and describe different types of Real-Time Control Strategies.

2.1. Evaluating PT Reliability

Reliability problems are most prevalent in complex PT systems with high demand. It is possible to divide the causes of reliability problems into two classes: (a) Internal causes, including factors such as driver behavior, passenger boarding and alighting at stops, improper scheduling, route configuration or inter-bus effects, which represent persistent problems, and; (b) External causes which are, by definition, more chaotic and these include traffic congestion and accidents, mechanical and technical failures and adverse weather conditions. The (a) persistent problems are addressed using (1) **Operational Planning (OP)** strategies, while the (b) recurrent but non-systematic problems are mitigated by (2) **Operational Control** [46].

It is possible to make a distinction between three aspects in evaluating PT reliability: (i) the unexpected increase in passenger waiting time; (ii) the time spent in crowded situations caused by on-board crowding, and; (iii) delays in passengers arrival times due to Travel Time Variability. The first two (i-ii) affect passengers comfort and travel experience, while the latter one (iii) disturbs passengers' daily activities [72] .

PT reliability could be measured either from bus vehicle or passenger points of view. There is a wide range of metrics for measuring service regularity [10]. Vehicle-based metrics may either refer to the discrepancy from the planned headway or correspond to a statistical measure of headway variability. The former could be defined in terms of the Headway Ratio. Let $f_{i,j}$ be the planned headway established between a given pair of trips, (i, j), and $H_{i,j}^b$ denotes the observed headway on this pair of trips at bus stop, b.

Table 1: Notation and symbols about the BB Control Framework employed.

n	total number of trips on the dataset of a given route
$f_{i,j}$	Planned Headway established for a given pair of trips, (i,j)
$H_{i,j}^b$	Observed Headway on a given pair of trips (i,j) at a bus stop b
b_i	i_{th} bus stop of a given route
T_i^j	arrival time of the bus running the trip i to the bus stop j of a given route
$TT_{(i,j)}$	travel time between two bus stops of interest $b_i, b_j : j > i$
$RT_i^{(l,l+1)}$	non-stop run time of the trip i in the road segment between two consecutive stops b_l, b_{l+1}
dwT_i^l	dwell time of a given vehicle/trip i on the bus stop b_l
s	total number of stops of a given route
η	headway-based minimum threshold to consider a BB event between two trips
$LTT_i^{(l,l+1)}$	Link Travel Time between two consecutive stops b_l, b_{l+1}
θ	number of days employed to build the training set to predict the LTTs on a daily basis
Δ_{y_i}	online update made to the previous LTT prediction y_i in place
r_y	residuals of the predictions made to y
α	constant user-defined learning rate of Δ_{y_i}
$\alpha(r_y)$	dynamic residual-based learning rate of Δ_{y_i}
κ^2	constant user-defined learning rate of $\alpha(r_y)$
e	the index of the most recently completed trip
P_e	set of LTT predictions made for the trip e
μ_e	average prediction residual for the completed trip e
ϕ	user-defined maximum threshold for the amount of trip-based residual μ_e
β^2	user-defined residual-based learning rate of $\alpha(r)$ to apply the trip-based update rule
E_c	offline prediction for the headways of the current trip c
$\mathbb{E}_c$	online prediction for the headways of the current trip c
HR_c	residuals of the headways' offline prediction for the current trip c
$H'R_c$	residuals of the headways' online prediction for the current trip c
$\omega(a,z)^i$	dynamic residual-based learning rate for the stop b_i given the headway's residuals a, z
$[\omega_{min}, \omega_{max}]$	user-defined parameters to bound the domain for the learning rate $\omega(HR_c, HR_c')$
D^{b_i}	Gaussian p.d.f. of the Headway between two consecutive trips on a given bus stop i
μ_{b_i}	mean value for defining the normal distribution D^{b_i}
σ_{b_i}	Standard deviation for defining the normal distribution D^{b_i}
τ	user-defined sliding window size to compute the recent Variance of $H_{k,k+1}^i$
$p(BB_{k,k+1}^i)$	BB likelihood for the pair of trips $\{k, k+1\}$ on the bus stop b_i
$\mathbb{D}^{b_j}$	descendent ordered vector of BB likelihoods for the downstream stops of b_j
BS^{b_j}	BB score to quantify the likelihood of occurring a BB event on the downstream stops of b_j
n^j	number of *agreements* (i.e. positive likelihoods) needed to compute BS^{b_j}
ψ	frequency-based threshold to trigger a BB alarm on stop B_j given BS^{b_j}
ρ	user-defined number of discrete bins employed to calculate ψ
b_ν	bus stop for which the BB event is predicted to occur
act^j	cor. action to be deployed once a BB alarm is triggered on b_j for the downstream stops
χ	symmetric user-defined min. threshold for the BB likelihood required to deploy a cor. action
χ_{BH}	min. threshold for the BB likelihood required to deploy Bus Holding
χ_{SS}	min. threshold for the BB likelihood required to deploy Stop Skipping
HT_k	Total Bus Holding Time to deploy to trip k
HT_k^i	Bus Holding Time to deploy to trip k on a given bus stop b_i
ζ_0, ζ	user-defined boundaries for the Total Bus Holding Time HT_k (in seconds)
K	Kernel function used to train the Support Vector Regression models
ϵ	Bandwidth, Hyperparameter of Support Vector Regression
γ	Hyperplane's Width, Hyperparameter of Support Vector Regression
d	Polynomial Degree, Hyperparameter of Support Vector Regression
C	Cost, Hyperparameter of Support Vector Regression
ς	Number of Terms, Hyperparameter of Project Pursuit Regression
ϖ	Number of Neighbors used in RReliefF algorithm

Table 2: Notation and symbols about the Simulations and Passenger Demand Model employed along this paper.

RTV	Run Time Variation on a given route
SAT	Scheduled Arrival Time of a given trip
AAT	Actual Arrival Time of a given trip
AWT	Average Waiting Time for the passengers of the trips running a given route
$AIVT$	Average In-Vehicle Time for the passengers of the trips running a given route
B^k	total number of passenger boardings on a route during a given trip k
$PAV_{z,k}^{b_j}$	the arrival time of the passenger z to b_j immediately before trip's vehicle k arrival at b_j
$bs_{z,k}/as_{z,k}$	boarding/alighting stop of a given passenger z on trip k
bo_k^i/al_k^i	number of boardings/alightings of the trip k on the bus stop b_i
o_{max}	maximum passenger capacity of a given bus vehicle
o_k^i	occupancy of the bus k after the boardings/alightings at bus stop b_i
υ	user-defined frequency's percentage to be used on calculating the passenger arrivals
$de_{k,k+1}^i$	num. of passengers arrived to the stop b_i during the headway between k and $k+1$
$\lambda_{min}, \lambda_{max}$	user-defined minimum/maximum threshold for the value of $\lambda(k)$
df_i	descendent demand factor of the bus stop b_i
φ	expected percentage of the route completed by any passenger on a given trip
$ns_{z,k}$	number of stops traversed by a given passenger z during trip k
$fas(z,i,k)$	function that determines whether the passenger z alighted on the stop i during the trip k
$SIM2, SIM1$	simulations run by deploying/not deploying automatic corrective actions
$\Delta_{BH,g}^j, \Delta_{SS,g}^j$	variation on T_g^j provoked by deploying a cor. action (Bus Holding or Stop Skipping) on a previously departed trip k
ξ	user-defined constant boarding time per passenger
dwT_{min}, dwT_{max}	user-defined boundaries for the dwell time
Δbo_k^g	variation on the boardings of the trip k on a stop b_g imposed by a given corrective action
$\mathbb{T}_k^g$	arrival time of the trip k to the stop g of a given route affected by a corrective action

Headway Ratio, i.e. $Hr_{i,j}^b$ is defined as follows [64, 65]:

$$Hr_{i,j}^b = (H_{i,j}^b / f_{i,j}^b) \times 100 \tag{1}$$

where the value 100 corresponds to a perfect headway adherence. In addition, for a given set of n trips, it is possible to compute the Standard Deviation and the Mean value of Hr (σ_{Hr}^b and μ_{Hr}^b, respectively) by calculating every possible $Hr_{i,i+1} : i \in \{1, ..., n-1\}$ at bus stop b. Then, the **Coefficient of Variation of Headway Ratio (CVHR)** at bus stop b is obtained as follows [12]:

$$HV^b = \sigma_{Hr}^b / \mu_{Hr}^b \tag{2}$$

These measures refer to vehicle arrival regularity but do not convey information on its implications on passengers experience. An irregular service implies an unexpected increase in passenger waiting times. The latter could be measured in terms of **Average Waiting Time (AWT)**. Let $PAV_{z,k}^{b_j}$ be

the arrival time of passenger z to bus stop b_j of a given route immediately before the vehicle performing trip k arrives at b_j. Then, it is possible to compute AWT of a route with s bus stops as follows

$$AWT = \frac{1}{\mathbb{B}} \sum_{k=1}^{T} \sum_{z=1}^{B^k} \sum_{i=1}^{s} \left(AAT_k^{b_i} - PAV_{z,k}^{b_i} \right) \tag{3}$$

$$\mathbb{B} = \sum_{k=1}^{T} B^k \tag{4}$$

$$B^k = \sum_{i=1}^{s} bo_k^i \tag{5}$$

where B^k stands for the total number of passenger boardings on a route for a given trip k, bo_k^i denotes the boardings on a given bus stop b_i on trip k and $AAT_k^{b_i}$ is the measured arrival time of trip k to bus stop b_i. This metric allows to directly assess the impact of alternative operational control measures on passenger waiting times. However, such measures (e.g. holding a bus in order to coordinate transfers) may also induce longer in-vehicle times [11]. In order to evaluate the overall impact on passenger travel times, the **Average In-Vehicle Time (AIVT)** should also be considered. Let $bs_{z,k}$ be the *boarding stop* of passenger z on trip k while $as_{z,k}$ is the respective alighting stop. The $AIVT$ can be computed as

$$AIVT = \frac{1}{\mathbb{B}} \sum_{k=1}^{T} \sum_{z=1}^{B^k} \left(AAT_k^{as_{z,k}} - AAT_k^{bs_{z,k}} \right) \tag{6}$$

Value-of-time studies concluded that waiting time is valued by passengers twice as much as in-vehicle time due to the discomfort and uncertainty it evokes [73]. Combined, these metrics reflect operators' and passengers' perspectives of PT reliability. Service providers address reliability problems by a combination of tactical and operational measures. At the tactical level, timetables are adjusted based on **Travel Time Prediction (TTP)** whilst at the operational level **Real-Time Control Strategies** are deployed. The following sections elaborate on these notions.

2.2. Travel Time Prediction

Let $TT_{(i,j)}$ be the run time between two bus stops $b_i, b_j : j > i$. It is possible to compute TT as follows.

$$TT_{(i,j)} = \sum_{l=i}^{j-1} \left(dwT^l + RT^{(l,l+1)} \right) \tag{7}$$

where $RT^{(l,l+1)}$ is the running time between two consecutive bus stops - the time interval between departure from stop b_l and arrival at stop b_{l+1} - and dwT^l is the dwell time at bus stop b_l.

A large range of methods are employed to Travel Time Prediction (TTP) problems from different research areas. Such approaches can be folded into four categories [8]: (a) ML and Regression Methods [19, 42]; (b) State-Based and Time Series models [57]; (c) Traffic Theory-based models [25] and (d) Historical data-based models [9, 14].

It is possible to differentiate short- and long-term TTP problems according to the prediction horizon considered. It is common to define such threshold to be between 60 to 180 minutes [38]. The long-term TTP is most commonly used for Planning - while the short term TTP is mainly used for Control and Operations (which is the scope of this paper). Despite being highly reliable when fed with *sufficient* amounts of data, long-term TTP predictions disregard the stochasticity of the network behavior as their predictive models are unable to **adapt** to exceptional demand events - such as traffic jams, accidents or major demand generators.

Recently, **hybrid** methodologies using both regression and state-based learning models are being proposed for short-term TTP [18, 75, 45] to overcome such limitations. The idea is to build long term TTP based on historical data which are being refined throughout the day using real-time information. These predictions can provide useful information to afford effective decision support on applying the best control strategy in each scenario in order to maintain service functionality. The following Section discusses alternative control strategies.

2.3. Real-Time Control Strategies

Control Strategies refer to the real-time implementation of service interventions. The goal is to restore service functionality when discrepancies occur. These strategies could take place at stops (e.g. holding, limited boarding) or between stops (e.g. speed adjustments, transit signal priority, stop skipping) [48]. These corrective measures are taken in order to mitigate sources of uncertainty which are most prevalent on high-frequency lines.

The deployment of **Holding** and **Stop Skipping** strategies is the focus of this study. Holding strategies are the most common control strategy for regulating bus services. The common practice is to instruct drivers to hold until the scheduled time at pre-defined stops (denominated time point stops)

in case of early arrivals. This schedule-based holding strategy helps maintaining the schedule and avoid early departures at scheduled transfers but is not effective for high-frequency services where headway variation is the main determinant of passenger waiting times. Pioneering PT systems have recently deployed headway-based holding strategies [10].

Holding strategies are an effective measure to improve service reliability. However, this is achieved by increasing passenger in-vehicle time and slowing down fleet operations. It is therefore desirable to combine it with a strategy that enables buses to shorten travel times. Stop skipping may be therefore deployed in extreme cases to allow drivers to skip one or several stops along the route in order to retain service reliability. This requires informing passengers on-board that intend to alight at the skipped stop. Note that both holding and stop skipping induce additional travel times to certain passenger groups in order to improve the overall service regularity . It is therefore necessary to conduct an adequate evaluation of the impacts of such strategies and deploy them only when they are expected to yield net benefits.

3. Data Preparation

The real-time framework for detecting and preventing bus bunching is applied to a case study system operated by STCP (*Sociedade de Transportes Colectivos do Porto*), the main mass public transportation company in Porto, Portugal. The public transport system in Porto consists of light rail and buses. The STCP operates 51 bus lines.

3.1. Data Collection

STCP's backoffice contains a two-nodes cluster which assures (1) the vehicles-to-infrastructure communication, and; (2) data storage. They were configured using a *hot/cold* server setup which assures the normal functioning of the system in case of failure. The AVL system consists of GPS-antennas installed on every vehicle plus 2G/3G antennas which enable real-time data interchanges between the vehicles and the central servers. Their communicational protocol is enabled by a General Packet Radio Service (GPRS), which broadcasts the vehicle's positions into the server's database every 30 seconds. These data packets are then stored in an Oracle relational database (i.e. 10gR2 [52]).

To construct this study's dataset, PL/SQL queries [28] were written to extract the datasets of a heterogeneous set of **nine bus lines** (A-I) - the set

includes both urban and non-urban routes covering different parts of great Porto area. The data was collected during a one year-period from January to December 2010. Each line has two route-directions A1, A2, B1, ..., I2. Additional information about network topology and the selected routes is included in Annexes (Section A.1.1).

3.2. Preprocessing

The studied fleet is equipped with differential GPS devices able to communicate each vehicle's position to the AVL data server. However, to obtain the trip data is not a trivial process due the absence of a *primary key* for identifying each trip individually in the server's database . It is thus necessary to (1) sort the data to then (2) match pairs of trips' arrival/departure in order to obtain link travel times . The data was sorted using the timestamps of vehicle's location associated to each link. The pairs were matched by identifying the records containing each trip' arrivals/departures with the planned schedule (which had the bus stops order and the scheduled arrival times for some of these stops - i.e. time point stops). Based on this information, it is possible to construct route datasets. Each dataset has one entry for each trip containing the following information: starting date of the trip, bus vehicle model, Driver ID, day of the year, type of day (normal day, holiday and floating holidays), departure time from a given bus stop and a stop ID.

As part of the preprocessing phase, the raw route datasets were processed in order to make it suitable for later stages. The final route datasets have one entry for each stop visit. Service features were selected based on visual inspection and the advice of transit experts, following a simplified version of the procedure conducted by Mendes-Moreira et al. [42]. The link travel time between each pair of consecutive stops is considered a random variable which varies as a function of eight independent variables (i.e. factors): (1) date (mapped as an incremental sequence starting with 1 for 01/01/2010), (2,3) weekday (weekdays - MON to SUN, and a day type - 1 for normal working days, 0 for other day types i.e.: holidays and strikes), (4) departure timestamp (in seconds), (5) destination stop id (categorical) and (6) origin one (categorical). In addition, each data record also has a (7) Trip ID (categorical) which identifies that a trip occuring on a shift of a particular day. Finally, the (8) day shift (categorical) is also included in our modelling approach.

Some data on link travel times is missing in the dataset (roughly 10% of the total information - see Table 4) due to the lack of pair matching and/or

other communication failures. To overcome this issue, the links for which there is no information are not considered in our predictive framework[1].

3.3. Dataset Statistics and Analysis

A feature analysis was conducted a priori to assess the predictive power of each feature. One of the algorithms used for this purpose is `RReliefF` [58]. It was already proposed for preprocessing purposes for this particular application by Mendes-Moreira et al. [42]. This is a filter-type of feature selection algorithm which outputs feature's weight with respect to the target variable based on its ability to reduce the entropy with small variations of its own value (i.e. neighborhood concept). 0.1% of each route's entries were selected to define such neighbors (i.e. $\varpi = {}^n/1000$). Table A.7 describes the results of such an evaluation test, summarized in Table 3. Additionally, a pairwise scatterplot of all the variables was used for visual inspection of the existing relations. For simplicity purposes, a randomly selected sample (i.e. 0.1%) of the most interesting routes (i.e. G1) was used to do it so. The resulting scatterplot is displayed in Fig. A.6.

The `RReliefF` illustrates that there is not a single variable which yields an effective predictive power. The only one which exhibits very low correlations with the target variable is the date, albeit with a considerable variation from route to route. Moreover, the pairwise scatterplots also do not uncover any relevant correlation among the features. The absence of evident correlations among the input data highlights the difficulty involved with the predictive task hereby carried out.

Table 4 presents brief descriptive statistics of the dataset. The columns correspond to each of the routes included in the case study. The rows show the following values per route: (1) total number of trips; (2) number of stops; (3) percentage of missing data for link travel times; (4,5) the maximum/minimum number of trips per day; (6,7) the maximum/minimum planned headway; (8) average measured headway during Peak Hours (PH); (9) total number of trips with a headway shorter than 25% of the planned headway at least once along their trip; (10) the average position where a BB took place along the route (e.g. 50% means that on average BB events took

---

[1] The predictions are done and updated as for any other link. However, as it is not possible to obtain the prediction residuals for these stops, they will not be considered when updating the predictive model regarding the headway at downstream stops. The reader is referred to section 4 for further information about this particular issue.

Table 3: `RReliefF` summary using an n-weighted mean±std.dev drawn from all routes.

FEATURE	RReliefF weights
(1) **DATE**	-0.386 ± 0.046
(2) **WEEKDAY**	-0.070 ± 0.028
(3) **HOLIDAY**	-0.007 ± 0.008
(4) **TIMESTAMP**	-0.010 ± 0.035
(5) **START_ST**	-0.002 ± 0.005
(6) **END_ST**	-0.001 ± 0.004
(7) **TRIP**	-0.010 ± 0.087
(8) **SHIFT**	-0.054 ± 0.032

place at a stop situated halfway along the route). The headway distributions of eight routes are also presented in Figure A.7.

The routes with the largest number of trips also exhibit the largest percentage of missing data (except for line D). The minimum planned headway among these routes is 9 minutes. Nevertheless, it is evident that headways vary considerably (as can be observed in Figure A.7). In particular, the thick left and right tails of the headway distributions, are especially pronounced for routes B1, C1, D2 and G1. These lines are characterized by short headways. Such tails illustrate that these routes often exhibit headways which are significantly shorter or longer than the scheduled ones. As a result, these routes exhibit the highest share of trips containing extremely short headways of less than 25% of the planned headway (between 3% and 7% of all stop-visits). While all routes are subject to headway variations, the extent of these variations vary among the case study lines due to differences in the underlying traffic conditions and demand profiles.

4. Methodology

The occurrence of a BB event is triggered by stochastic processes and hence difficult to predict. Notwithstanding, current system states may allow uncovering such future occurrences. To do so, it is not sufficient to mine historical AVL data as there is no obvious trend or a simple *static* association rule which can explain such events. In [43], it is postulated that a BB event is usually preceded by a headway deviation prevailing further upstream along the route. However, such a rule cannot handle the random series of events that may affect a given bus trip which can arise sporadically rather than

15

Table 4: Descriptive statistics for each route considered. Headways in minutes.

	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2
Num. of Trips	10108	10224	24554	24388	20598	20750	25862	25674	18651	18940
Nr. of Stops	18	18	30	30	26	26	22	22	26	26
% of missing data	0.11%	0.09%	5.05%	4.93%	11.44%	6.70%	3.55%	1.20%	8.01%	4.98%
Min. Daily Trips	28	28	50	51	44	45	62	63	51	63
Max. Daily Trips	45	44	98	97	76	76	95	91	67	91
Min. Headway	20	18	10	9	10	11	9	9	14	14
Max. Headway	120	120	61	66	100	92	62	72	111	111
Avg. PH Headway	28.53	30.99	19.81	19.92	16.01	15.64	14.91	14.49	21.35	16.52
Trips w/ BB	19	43	734	811	682	553	1009	885	291	211
% of Bunching	0.18%	0.42%	2.99%	3.33%	3.31%	2.66%	3.90%	3.45%	1.56%	1.11%
BB Avg. Pos.	53.78%	82.94%	58.41%	76.28%	63.22%	74.87%	60.19%	62.13%	53.51%	67.89%
	F1	**F2**	**G1**	**G2**	**H1**	**H2**	**I1**	**I2**		
Num. of Trips	20054	19361	26739	26007	11319	11864	15691	14901		
Nr. of Stops	32	32	45	45	31	31	24	24		
% of missing data	4.34%	2.17%	10.74%	7.5%	0.25%	0.47%	2.25%	7.23%		
Min. Daily Trips	56	57	65	71	29	29	35	35		
Max. Daily Trips	85	84	100	101	39	42	59	54		
Min. Headway	12	13	10	10	20	19	17	19		
Max. Headway	112	120	60	101	120	120	120	120		
Avg. PH Headway	24.31	24.81	14.44	13.92	31.01	30.65	23.82	22.15		
Trips w/ BB	437	364	1917	1702	17	23	388	225		
% of Bunching	2.18%	1.88%	7.17%	6.54%	0.15%	0.19%	2.47%	1.51%		
BB Avg. Pos.	58.32%	68.55%	49.71%	53.63%	56.57%	52.75%	60.79%	69.70		

systemically [66]. Consequently, an *off-the-shelf* ML method will not be sufficient to handle this specific problem.

This section describes the details of a stepwise learning methodology to detect and then prevent the BB phenomenon from propagating in real-time. It utilizes simultaneously historical and real-time AVL data. The framework works on two different parts: (I) **BB Event Detection** and (II) **Corrective Action Deployment**. The first part is an Advanced ML framework comprises of the following three steps:

- (I-1) an offline regression method is used to predict **Link Travel Times** (i.e. the time interval between the arrival times at two consecutive bus stops) for every trip in the following day (the forecasting horizon) using some of the most *recent* days (the learning period) to train our model;

- (I-2) these predictions are constantly refined (i.e. online learning) using (I-2a) trip-level information as well as (I-2b) stop-based information. Both steps are based on the **Perceptron Delta Rule** [60] by reusing each prediction's residuals to improve successive predictions. After

two consecutive trips of interest depart from their origin stop, a BB monitoring framework is triggered;

- (I-3) this framework estimates the **likelihood** of a BB event to occur at downstream stops by assuming that headway is normally distributed.

Given a certain user-defined threshold, a BB detection alarm is *launched*. The second part consists of the following two steps:

- (II-4) selecting one out of two possible **corrective actions** (Bus Holding or Stop Skipping) based on the relative headway deviation;

- (II-5) finally, the exact details of action implementation are specified based on service conditions and the designed set of corrective actions.

Parts I and II of the methodological framework are illustrated in Figures 2 and 3, respectively. It is formally presented on the following sub-sections (where steps I-1, I-2 and I-3 follow closely sections 3 and 4 in [45]).

4.1. Step I-1: Link Travel Time Prediction

Let trip i of a given bus route be defined by $T_i = \{T_i^1, T_i^2, ..., T_i^s\}$ where T_i^j stands for the arrival time of trip i at bus stop j and s denotes the number of bus stops along the route.

Consequently, the *observed* headways between two buses at stop j running on consecutive trips $k, k+1$ is defined as follows

$$H_{k,k+1} = \{H_{k,k+1}^1, H_{k,k+1}^2, ..., H_{k,k+1}^s\} : H_{k,k+1}^j = T_{k+1}^j - T_k^j \qquad (8)$$

Under optimal conditions, the headway between two consecutive trips is expected to be *constant* (i.e. $H_{k,k+1}^i \simeq f_{k,k+1}, \forall i, k$). However, in reality bus services are subject to uncertainty that results with service variability as discussed above. A BB event is expected to occur when headways become **unstable** until eventually forming a platoon. The operational control framework proposed in this paper calls for the deployment of corrective actions when the headway deviates beyond from the planned headway a certain threshold (i.e. η). The value[2] of η is specified as a function of the planned headway, i.e. $f_{k,k+1}$.

---

[2] to know more about the BB event labelling process, the user should consult the Section 2 in [45]
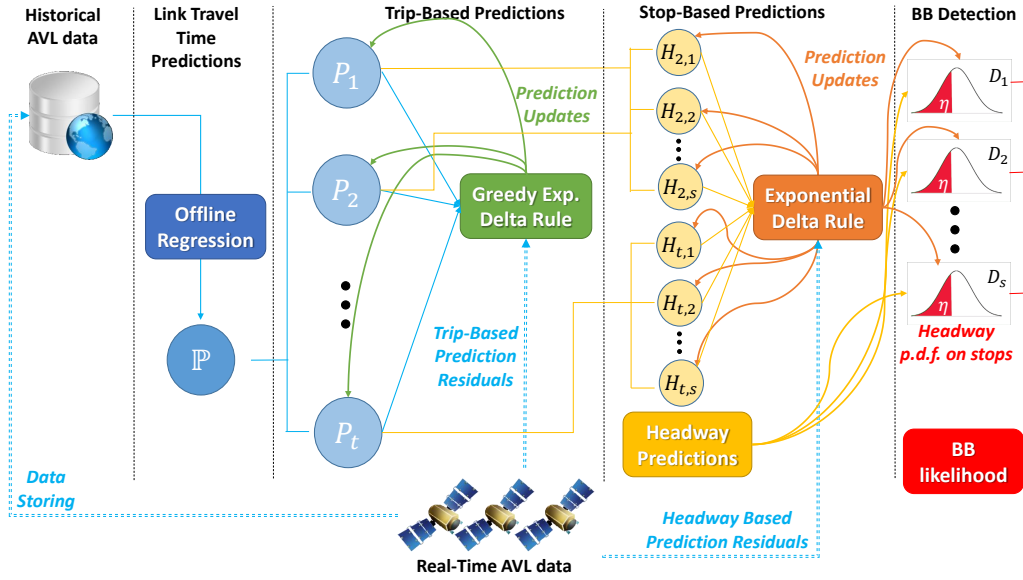
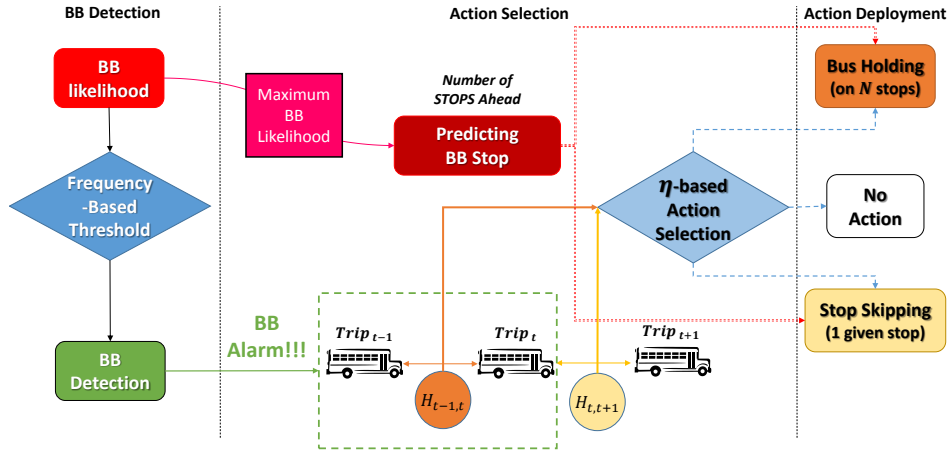Figure 2: Illustration on the BB detection framework (part I).

Figure 3: Illustration on the BB detection framework (part II).

Consequently, it is possible to devise a *recursive* relationship between BB occurrences, the observed headway and the arrival time of a given trip i to a bus stop of interest l, i.e. T_i^l. Let the arrival time and the Link Travel Time be defined as follows

$$T_i^{l+1} = T_i^l + dwT_i^l + RT_i^{(l,l+1)} \tag{9}$$

$$LTT_i^{(l,l+1)} = dwT_i^l + RT_i^{(l,l+1)} \tag{10}$$

By observing eqs. (8,9,10), it is possible to infer the following recursive relationship between headways measured on consecutive bus stops

$$H_{k,k+1}^{l+1} = H_{k,k+1}^l + LTT_{k+1}^{(l,l+1)} - LTT_k^{(l,l+1)} \tag{11}$$

Logically, it is possible to infer the future values of $H_{k,k+1}$ based on the predictions on the future values of LTT. Hereby, we propose to perform **long-term TTP** based on the dataset described in section 3 in order to approximate the headways between every pair of consecutive trips. Departing from the work of [42], it is possible to formulate the Link Travel Time prediction problem as an inductive learning regression problem. It involves inferring the following function

$$\bar{f} : X \to \mathbb{R} : \bar{f}(x) = f(x), \forall x \in X \tag{12}$$

where X stands for the *feature* set and f represents the unknown explanatory function. A dynamic training set is used by employing **windowing** [74]: just the most recent data (i.e. LTT for the latest θ days, where θ is a user-defined parameter) is considered to train a $\bar{f}$ able to predict the LTT values for all the trips that take place on the following day.

By doing so, we aim to obtain an optimal fit of $\bar{f}(x)$ for a given training set (i.e. $\bar{f}(x) = f(x)$). This type of learning tasks is often denominated offline learning. Attaining an *optimal* solution is one of the key characteristics of this type of models. However, it can also be regarded as its major drawback because it is unable to adjust itself to changes introduced in the process by stochastic events (as discussed in section 2.2).

With this aim, we propose a **hybrid learning** model - which *combines* offline learning and online learning models. The offline regression produces a *context free* prediction for the LTT distribution throughout the day on a given route while the online learning handles the constant drifts that the learning process introduces due to multiple stochastic events that arise during system operations. Such an online learning task involves updating these predictions based on the residuals produced by earlier predictions. The residual-based update procedure is described in the following section.

4.2. Step I-2: Delta Rule as a Residual-Based Update

One of the most well-known offline learning techniques for regression are Artificial Neural Networks (ANN) [4]. ANNs are computational models inspired by the neuron's brain structure. Perceptrons are the basic component of an ANN [60]. They play the role of a neuron in human brain. Typically, a Perceptron receives a set of inputs and computes their weighted sum. The output of the Perceptron is computed by an activation function, e.g. sigmoid, of the weighted sum of the inputs. Learning in a Perceptron consists of finding the weights, typically using (Stochastic) Gradient Descent, that minimizes a loss function of interest (e.g. l_2 loss - the squared difference between the outputs and real values) . In ANNs, Perceptrons are organized in layers, where the output of one layer acts as input to the next layer. ANNs have been used in TTP, for example in Chien et al. [19], Chen et al. [18], Jeong and Rilett [37], Mazloumi et al. [40].

The most well-known type of ANNs is a **Feedforward Neural Network (FNN)** - where the information just moves forward, from the input nodes to the output nodes. The Backpropagation Algorithm [41] is a State-of-The-Art method to train those networks . It progresses the outputs of their nodes *forward* while the residuals are propagated *backwards* to update the network weights until a convergence criterion is met (e.g. the average residual is below a given threshold). Such a learning task is performed by employing the **Delta Rule (DR)** [63]. Let w_{ji} be the weight of the link connecting the i_{th} input node (with an input value of x_i) to the j_{th} output node where y_j is the node's current output and t_j is the target output. The delta rule updates the weight by adding to the previous weight w_{ji} a given Δ_{ji} as follows

$$w'_{ji} = w_{ji} + \Delta_{ji}; \Delta_{ji} = \alpha(t_j - y_j)x_i \tag{13}$$

where α stands for an user-defined parameter (i.e. typically, $1 >> \alpha > 0$). By running through multiple iterations, this algorithm will *force* the weights to *converge* in order to find a *local minimum* of a function (i.e. to aproximate $\bar{f}(x)$ and $f(x)$). Obviously, a reasonable knowledge of the problem is normally required to perform an adequate feature selection and parameter setting such as the number of hidden layers and learning rate α in order to successfully apply ANNs to TTP problems [3]. Moreover, ANNs also require a comprehensive amount of data and computational power to allow the learning model **absorb** all the dependencies between the input and the output values. Nevertheless, they are not able to adapt themselves to

20

handle *unseen* concepts and drift their output accordingly. However, when we are facing a large-scale data stream of information - such as the LTT dataset communicated by each vehicle – we face an high latency source of spatiotemporal data. Is it possible to turn this high information latency into an *advantage* by employing an ANN-based learning?

To pursue this idea, we propose to adapt the delta rule to **incrementally update** the predictions firstly obtained by the offline regression learning process. We consequently modified eq. 13 as follows

$$y_i' = y_i \times \Delta_{y_i}; \Delta_{y_i} = \alpha(t_{i-1} - y_{i-1}) \tag{14}$$

turning it into a first-order update rule where the next prediction y_i is updated as soon as the real value of the previous one (i.e. t_{i-1}) is known. It thus consists of adding a percentage of the residual of y_{i-1}, i.e. $r_{y_{i-1}} = (t_{i-1} - y_{i-1})$ to the explantory model's output. The basic idea is that the learning model will not change dramatically within several hours (i.e. one day) but it will instead drift gradually in response to changes in the expected values (i.e. a large-scale traffic jam). As these types of phenomena are also temporary, they also need to be forgotten as soon as they have terminated (i.e. in response to a progressive decrease of the $r_{y_{i-1}}$ value). We denominate this update Linear Delta Rule. This algorithm was successfully deployed for other online learning problems (see, for instance, the incremental computation of the ARIMA weights in [44]).

To improve the model ability to **react**, the learning rate α may also be updated based on the progression of residual's value (i.e. $\alpha(r_y)$). Such update can be computed as follows

$$\alpha(r_y)' = \alpha(r_y) \times (1 + \vartheta \times (1 + \kappa^2)) \tag{15}$$

where κ^2 sets the progression rate of $\alpha(r_y)$ and ϑ stands for the number of *consecutive* residual's with the same signal (i.e. positive/negative). Consequently, κ^2 is a *quadratic* learning rate (i.e. user-defined) that sets the rate on how the original learning rate α is updated, while ϑ is a variable that refers to trending in our prediction (i.e. over/under estimation). For a given prediction i, it is computed recursively as follows

$$\vartheta_i = \begin{cases} \vartheta_{i-1} + 1 & \textbf{if } \frac{r_{y_{i-1}}}{r_{y_i}} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

21

where $\vartheta_0 = 0$. The variation of delta rule described in eqs. (15,16) is denominated **Exponential Delta Rule** (Exponential DR). This algorithm was also successfully deployed for other online learning problems (see, for instance, the incremental exponential adaptation of the interval sizes in [51]).

The Exponential DR is applicable whenever a more sensitive reaction to changes on the residuals is desirable as compared to the Linear DR which is a more generic first-order update rule. Sometimes, the Exponential DR can also be applied directly to the output value as the learning rate α is also learned from the residual's distribution (i.e. $\alpha(r_y)$). Consequently, it is possible to re-write eq. 15 as follows

$$\Delta_{y_i} = \alpha(r_y) \times y_{i-1} \tag{17}$$

which gives an even greater reactivity to the learning model [51]. This model is therefore named **Greedy Exponential DR**.

In the following couple of subsections, we detail the application of these rules to incrementally update the LTT predictions.

4.3. Step I-2a: Trip-Based Link Travel Time Update

Let e denote the last trip completed before the current trip c started. The trip-based refinement compares the predictions of e, i.e. $P_e = \{P_e^1, P_e^2, ..., P_e^s\}$ with their real values T_e in order to update the value of P_c. First, we compute the residuals as $r_e = T_e - P_e$ and then its average value (i.e. r_P) as follows

$$r_e = \mu_e = \sum_{i=1}^{s} \frac{r_e^{\,i}}{s} \tag{18}$$

Second, a user-defined percent-wise maximum threshold $0 < \phi << 1$ is employed to identify trips with an error larger than expected as $\phi \times f_{c,e}$. Then, a Greedy Exponential DR is employed to update the LTT predictions for the next trip P_c as follows

$$P_c' = P_c + \Delta_{P_c}; \Delta_{P_c} = (\alpha(r_P) \times P_c) \tag{19}$$

where the dynamic learning rate $\alpha(r_P)$ is updated as follows

$$\alpha(r_P)' = \alpha(r_P) \times \left(1 + \vartheta \times (1 + \beta^2)\right) \tag{20}$$

where β^2 stands for the user-defined *quadratic* learning rate of the trip-based $\alpha(r_P)$. The threshold $\phi \times f_{c,e}$ is employed over the learning rate of $\alpha(r_P)$ by

constraining the progression rate of $\alpha(r_P)$ defined in eq. 16 as follows

$$\vartheta_c = \begin{cases} \vartheta_{c-1} + 1 & \textbf{if } \frac{\mu_e}{\mu_c} = 1 \wedge \mu_e > (\phi \times f_{c,e}) \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

These updates are performed **incrementally** (i.e. every time a link is traversed and the respective travel time becomes available). Note that the residuals are always calculated over the regression results P_c and not over the updated arrays P'_c. Hence, its computation is iterative but not recursive.

4.4. Step I-2b: Stop-Based Headway Update

Given the updated predictions of two consecutive trips P'_c, P'_{c+1}, it is possible to obtain the predicted headways $E_c = P'_c - P'_{c+1}$ while the actual headways are computed as $H_c = T_c - T_{c+1}$. The computation of E_c is performed offline since it does not use information about the current headway. The second refinement uses the headway residuals $HR_c = H_c - E_c$ to update E_c stop-by-stop. Incrementally, we obtain online headway predictions as $\mathbb{E}_c^i = H_c^{i-1} + E_c^i - E_c^{i-1}, \forall i \in \{2, s\}$. The problem is updating the headway prediction for the next stop (i.e. E_c^i) given the value of HR_c^{i-1}. To this end, we employ the Exponential DR, implemented with the following first-order rule:

$$E_c'^{\,i} = \mathbb{E}_c^i + \Delta_{\mathbb{E}_c^i}; \Delta_{\mathbb{E}_c^i} = \omega^i(HR_c, H'R_c) \times HR_c^{i-1} \quad (22)$$

The dynamic learning rate $\omega^i(HR_c, H'R_c)$ is updated as follows

$$\omega^i(HR_c, H'R_c)' = \begin{cases} \omega^{i-1} \times (1 - \omega^{i-1}) \text{ if } |HR_c^{i-1}| > |H'R_c^{i-1}|, \\ \omega^{i-1} \times (1 + \omega^{i-1}) \text{ otherwise.} \end{cases}$$
$$\text{subject to } \omega(HR_c, H'R_c^{i-1}) \in [\omega_{min}, \omega_{max}] \quad (23)$$

where $|HR_c|$ stands for the absolute residuals of the Headway's offline prediction E_c, $|H'R_c|$ denotes the absolute residuals of the Headway's online prediction $\mathbb{E}_c$ and $[\omega_{min}, \omega_{max}]$ stands for a user-defined boundary for the ω_r's range of values. Again, the entire headway array $\mathbb{E}_c^q, q \in \{i + 1, s\}$ is constantly updated with the most recent headway value H_c^i as soon as it becomes available. This scheme provides a certain flexibility to handle the real-time stochasticity associated with headways. By performing these two steps, it is possible to maintain distinct levels of information to approximate the real-time link travel times incrementally. The propagation of our updates to downstream stops along the trip is key in antecipating BB, as explained in the following section.

4.5. Step I-3: BB Event Detection

A probabilistic framework for detecting a BB event at downstream stops is proposed. The **likelihood** of a BB event to occur at any of the downstream stops between two consecutive trips $(k, k+1)$ is computed by inferring the short-term probability distribution function (p.d.f.) of their headways $H_{k,k+1}$ at each stop, i.e $D(H_{k,k+1}^{b_i}) \equiv D^{b_i}, \forall i \in \{1..s\}$.

Let $M_k^{b_i} = \{M_{k-1}^{b_i}, ..., M_{k-\tau}^{b_i}\} : M_j^{b_i} = \left| H_{c,c+1}^{b_i} - \mathbb{E}_{c,c+1}^{b_i} \right|$ denote an array containing the most recent τ residuals of the headway predictions made at bus stop b_i, where τ is an user-defined parameter that defines the short-term memory size. To calculate such a p.d.f., it is postulated that

Assumption 4.1. *The Headway p.d.f. at bus stop b_i, i.e. D^{b_i}, follows a* ***Gaussian distribution*** *defined as $D^{b_i} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i})$.*

where μ_{b_i} is the expected headway value defined by $\mu_{b_i} = \mathbb{E}_{c,i}$ while σ_{b_i} is given by the median value of the recent prediction residuals (i.e. $\tilde{M}^{b_i}$).

Considering the hypothesis of a BB event occurring at this specific stop, we can express its likelihood as $p(BB_{k,k+1}^i) = p(H_{k,k+1}^i \leq \eta \mid \mathbb{E}_c^i, M^{b_i})$. This definition allows to quantify the p-value of a BB event to occur at a certain stop. It is then possible to quantify a Bunching likelihood for all downstream stops (and also to update them each time we obtain a more up-to-date headway value).

The aforementioned assumption 4.1 might induce a certain error because the headway distribution on a given stop may follow, on some circumstances , different p.d.f. [35]. For simplicity and to allow its incremental computation, we consider all of the headway distributions to be Gaussian. To handle the error introduced by this assumption, a Bunching Score (BS^{b_j}) is estimated for a given bus stop b_j based on the estimations of headway distributions D^{b_i} between trips c and $c+1$ for downstream stops. Let $\mathbb{D}^{b_j} = \bigcup_{i=j+1}^{s} p(BB_{k,k+1}^i)$ be an ordered vector (descendent) containing the likelihoods for downstream bus stops. The BS can be obtained as follows:

$$BS^{b_j} = \frac{1}{n^j} \sum_{i=1}^{n^j} \mathbb{D}^{b_j}; n^j = \lceil 3 - ((j-1) \times 3/s) \rceil : n^j \in \mathbb{N} \qquad (24)$$

where n^j is the number of likelihoods used to compute BS^{b_j}. n^j determineshow many stops should set a positive alarm antecipating BB. Finally,

a *BB* is said to be likely to occur at stops downstream of b_j if it exceeds a threshold value, ψ, defined as follows

$$\psi(f_{c,c+1}) = 0.3 + [(f_{c,c+1} \mod \rho) * 0.1] : 0 < \psi(f_{c,c+1}) \leq 1 \qquad (25)$$

where ρ stands for a user-defined parameter for the number of threshold bins that should be employed.

By employing n^j when computing BS^{b_j}, the method strives for *consensus* by requiring high BB likelihoods for multiple bus stops whenever BB events are being predicted at an upstream segment of a given route (i.e. longer forecasting horizons). Such a consensus is also an way to handle the error introduced by assumption 4.1. After a BB alarm is triggered for a given bus stop b_j, a corrective action is implemented. In the next section, a process to automatically determine which action should be deployed in each situation is described.

4.6. Step II-4: Selecting a Corrective Action

In this study, two corrective actions are considered: (1) Bus Holding and (2) Stop-Skipping. These actions were selected because of their simplicity, easy communication and deployment [23]. Once a BB event between two consecutive trips $c, c+1$ is predicted for stops located downstream of b_j, it is essential to determine which control strategy should be deployed.

The procedure for strategy selection starts by determining which is the bus stop where the BB event is most likely to occur, i.e. b_ν - which is determined by selecting the stop that maximizes the BB likelihood:

$$b_\nu = \arg\max_{b_i} p(BB^i_{c,c+1}) : j < i \leq s \qquad (26)$$

Let $act^j \in \{0, 1, 2\}$ be the corrective action to be applied given that a BB alarm is triggered for bus stop b_j. The value 1 corresponds to deploying *Bus Holding*, 2 implies *Stop Skipping* and 0 does not involve any intervention. The value of act^j is selected based on the headways between the current trip

c, the previous one $c - 1$ and the following one $c + 1$:[3].

$$act^j = \begin{cases} 2 & \text{if } \chi_{BH} \leq p\left(H^{b_\nu}_{c,c+1} \leq \eta\right) \geq p\left(H^{b_\nu}_{c-1,c} \geq ((2 \times \sigma_{b_\nu}) - \eta)\right), \\ 1 & \text{if } p\left(H^{b_\nu}_{c,c+1} \leq \eta\right) < p\left(H^{b_\nu}_{c-1,c} \geq ((2 \times \sigma_{b_\nu}) - \eta)\right) \geq \chi_{SS}, \\ 0 & \text{otherwise.} \end{cases}$$

(27)

where χ_{BH} and χ_{SS} stand for two user-defined minimum thresholds for the BB likelihood required to deploy an action for a given PT system. The following step prescribes how the chosen strategy is implemented. The idea behind eq. 27 is to deploy stop skipping to address very particular situations, when we need not only to correct the headway between the current vehicle and its subsequent one, but also the one between the current vehicle and the following one. Such need arises in situations where this last pair is experiencing very long headways, as illustrated in Figure 4.

---

[3] note that the distributions' parameter values were omitted to simplify the equation's readability;
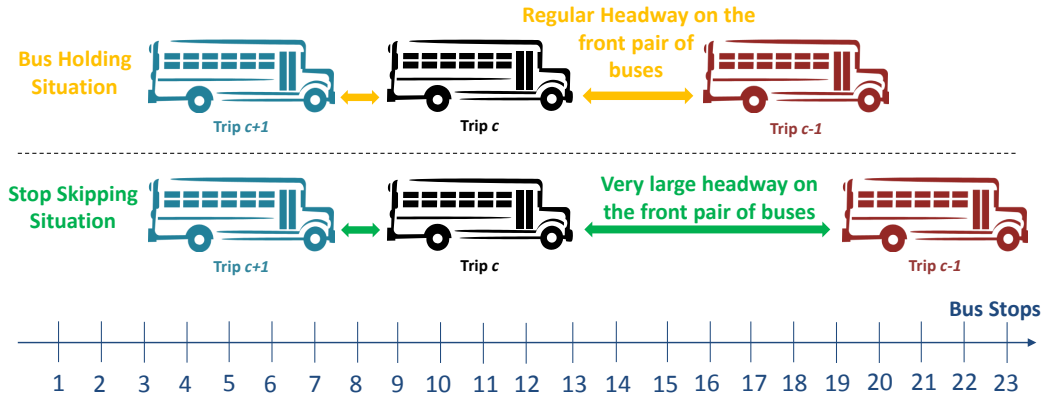

Figure 4: An illustration of correction action suitability – holding trip $c + 1$ at the next stop (top), trip c skipping the next stop (bottom). Trip $c/c + 1$ has to be slowed down or speeded up depending on subsequent headways.

4.7. Step II-5: Implementing a Corrective Action

Once selected, the implementation of the control strategy has to be specified. If $act^j = 1$, then the holding time for bus $c + 1$ is set as follows

$$HT_{c+1} = \eta - H^j_{c+1,c} + 10 : HT_{c+1} \in (\zeta_0 \times \{1,..,\zeta\}) \qquad (28)$$

where $\{\zeta_0, \zeta\}$ are user-defined boundaries for the Total Holding Time (in seconds) to realistically reflect the driver-communication system limitations [13]. Furthermore, an upper limit is set to the holding time per stop due to passengers' acceptability reasons. The Total Holding Time is therefore **distributed** over multiple stops. Therefore, the bus holding time at each stop, HT^i_k, is computed as follows

$$HT_{c+1} = \sum_{i=j+1}^{\nu} HT^i_{c+1} : HT^i_{c+1} \in (\zeta_0 \times \{1,..,\zeta\}) \wedge \left(HT^i_{c+1} - HT^{i+1}_{c+1}\right) \leq \zeta_0$$

$$(29)$$

If $act^j = 2$, the bus c is set to skip bus stop b_{j+1} or the subsequent stop if there is no possibility of informing passengers beforehand.

The feasibility of the abovementioned framework is investigated using a computer simulation which was fed with real-world data (described in Section 3). The details of these experiments along with their results are presented in the following Section.

5. Experiments

In this section we first describe the experimental setup employed. Second, a tuning framework is proposed to adjust the set of parameter used in this framework for any case study of interest. Then, the evaluation metrics employed in this work are described, along with the details of the passenger demand profile generation process (in Annexes). Finally, experiments' results are presented.

5.1. Experimental Setup

Throughout this work, the value η was set to $\eta = f_{k,k+1}/4$ following previous studies of this particular case study [43, 45]. For the offline regression problem, we also followed the experimental setup firstly proposed by Moreira-Matias et al. [45] in employing a training sliding window of $\theta = 7$ days. As a baseline offline regressor, we picked Random Forest (RF) [5], following the

insights proposed by Mendes-Moreira et al. [42]: "*RF ends up being the most interesting method from an off-the-shelf perspective for long-term travel time prediction. It presents competitive results with little preprocessing effort.*" The primary reason for selecting it is its ability to avoid overfitting, following a good convergence behavior for sensible default values available in most of its implementations [5, 29].

All of the experiments were conducted using the R Software [56]. R is a free software environment for statistical computing and graphics. In this context, we used it to conduct all the regression procedures detailed in Section 4, as well the simulation described in Section A.2.3. The implementation used for the offline regression procedure was drawn from the R package [randomForest]. One of its main drawbacks is that it does not support variables with more than 32 factors. To deal with this issue, each variable whose range exceeded this threshold was mapped into a N-multidimensional space where each variable represents a subset of the initial categorical range.

The learning framework proposed in this paper contains a total of ten parameters: $\{\beta^2, \phi, \omega^0$ (i.e. the first value of the learning rate)$, \omega_{min}, \omega_{max}, \tau, \rho, \chi_{BH}, \chi_{SS}, \zeta_0, \zeta\}$. A symmetric threshold was established in this case study for deploying corrective actions in this application (i.e. $\chi \equiv \chi_{BH} \equiv \chi_{SS}$). However, different values could be assigned to these parameters, depending on the operational plan devised for a given system [7].

It is possible to divide this set into two types of parameters: prediction parameters (the first seven) and the deploying parameters (the last three). The deployed parameters must be set by the agency due to their interplay with the Control policies already in place [10]. In this particular case, the parameters were set to $\{\chi = 0.5, \zeta_0 = 30$ (in seconds)$, \zeta = 4\}$. However, the prediction parameters require an adequate setting for minimizing errors. The tuning task is described below.

5.2. Parameter Tuning and Analysis

RF was run using the hyperparameters previously suggested by Moreira-Matias et al. [45] for this task: {mtry=3, ntrees=750}. The classical techniques for optimizing such hyperparameters (e.g. cross-validation) are often not easily applicable to sequential data whenever no validation set is available. Moroever, we assume statistical independence among routes, which forbiddes the usage of samples from an entire route (i.e. leave-one-out) for this purpose. In our case, the first seven days of January of 2010 are the only true training/validation set available - since the remaining days have already

being used for validating our methodology. This dataset does not contain a sufficient amount of data to carry out such an analysis. To overcome this limitation, we used parts of the test set by using the entire month of January[4]

The parameters $\{\beta^2, \phi, \omega^0, \omega_{min}, \omega_{max}, \tau, \rho\}$ can be divided into three different classes: 1) τ, ϕ are associated with the proposed drift detection mechanism while 2) $\beta^2, \omega^0, \omega_{min}, \omega_{max}$ define our learning rate. Finally, 3) ρ is relative to our BB detection framework.

In Change Detection problems, the major issue is often related to distinguishing between *noise* and *change* in data following non-stationary distributions [33]. Hereby, we propose a basic drift detection method with a fixed window size: τ. It expresses our expectation concerning the *seasonality* of the signal that is being processed (i.e. the distribution is expected to be stationary within such period). Conversely, ϕ denotes the admissable *noise* in our process, determining the threshold used to distinguish it from a *change* - which need to be handled differently.

Of the remaining class, $\beta^2, \omega^0, \omega_{min}, \omega_{max}$ *bound* the learning process to ensure that it will not diverge on any condition. Similarly to most common learners used among with neural network topologies (e.g. Stochastic Gradient Descent [61]), an adequate setting of their values is crucial. The variation induced to this supervised learning framework by changing one of these parameter values is not homogeneous (e.g. a variation on the learning rate β^2 will affect the output values more than a change in the value of τ). Following the previous experiments of the prediction task [45], the value of ω^0 was set to 0.1 on a daily basis (as this value just affects the first stop-based update). The value of ρ was set to $\rho = 360$ seconds [45].

The values of the remaining parameters must be tuned for each individual route since we assume that they are statistically independent. Indeed, this assumption is supported by empirical observations: the travel times on consecutive route segments do not exercise significant correlations [70]. To carry out such tuning, we employ a simplified version of the Sequential Monte Carlo method [6]. It consists of *randomly* sampling data from the training set regarding subsets of its feature space (typically, 10% to 30%) to determine the combination of parameter values which perform better, on average, for these samples. Ideally, the application of this framework to this problem

---

[4] Note that this procedure do not necessarily imply overfitting. This is adequately illustrated by the high variance uncovered by the RReliefF results in Table 3.

would consist on randomly selecting a certain number of individual days of historical data (e.g. last year) which could cover most of the possible cases (e.g. days from every month, every daytype). Similarly to the hyperparameter tuning task described above, we also used January as our *sandbox*, by randomly sampling ten days containing all weekdays and daytype. Notwithstanding, we stress that both of the abovementioned procedures should be ideally conducted using historical data to achieve a satisfactory tuning of the (hyper)parameter sets.

All possible combinations of the following parameter settings were considered in our experiments: $\phi = \{5\text{E}^{-3}, 1\text{E}^{-2}, 2\text{E}^{-2}, 5\text{E}^{-2}, 7.5\text{E}^{-3}, 0.1\}$, $(\omega_{min}, \omega_{max}) = \{(1\text{E}^{-3}, 0.3), (5\text{E}^{-3}, 0.3), (1\text{E}^{-2}, 0.3), (1\text{E}^{-3}, 0.4), (5\text{E}^{-3}, 0.4), (1\text{E}^{-2}, 0.4)\}$, $\beta^2 = \{1\text{E}^{-2}, 5\text{E}^{-2}, 1\text{E}^{-1}, 3\text{E}^{-1}, 5\text{E}^{-1}\}$ and $\tau = \{3, 5, 10, 15\}$. The obtained parameter setting is displayed in Table A.8.

5.3. Evaluation Metrics

The evaluation of our framework is divided into three dimensions: (i) the mean absolute error (MAE) of headway predictions, (ii) the BB detection accuracy and (iii) the effect of the actions deployed on PT users. The details are provided in the Annexes (Section A.2.2).

5.4. Demand Profile Generation

In the absence of empirical passenger counts, a synthetic demand profile was devised based on real-world LTT, the planned schedule and on a series of assumptions on passenger load accumulation. Passenger load was assumed to follow an accumulation pattern typical for radial services that traverse through central areas of the urban agglomeration. Passenger arrival is assumed to follow a Poisson process based on service headways and on-board occupancy adheres to flow conservation rules and enforces capacity constraints. The demand profile generation is detailed in Section A.2.3.

5.5. Results

Table 5 contains a summary of the prediction results (full results in Table A.10) while the effects of the corrective actions are introduced in Table 6.

6. Discussion

This Section presents a critical view of the methodology developed in this study. First, we depict the results obtained for the case study dataset. Then,

the limitations present in the State-of-The-Art that are addressed in the proposed methodology are discussed, along with the prospects for its deployment and the potential impact of this framework on a mass transit agency (in Annexes). Finally, the generality of this framework to approach any type of task where it is necessary to perform regression while still handling bursty and/or abrupt concept drifts is discussed. .

6.1. Result Analysis

The performance of the headway predictive framework varies from route to route - even for opposite directions of the same line. For instance, the offline prediction performance on line G is four times greater for one of the route-directions than for the other one. Notwithstanding, it is important to stress that in both cases the predictive framework performs reasonably well (i.e. an error of $\simeq 30$ sec). The online learning framework converges to the real output values by reducing the average error by more than 90% (i.e. with the exception of route I2).

The long BB prediction horizons (i.e. roughly 11 stops, on average), enable a gradual and incremental implementation of corrective actions. At the same time, the values of the precision are low - especially for certain routes (i.e. lines A and H) due to triggering more alarms than are actually necessary (i.e. false positives). However, this pattern is admissible in the context of this problem - since it is desired to deploy corrective actions in a **proactive** fashion. The routes with lower precision values are those characterized by lower frequencies. This suggests that the BB detection threshold values (i.e. χ and η) should not be uniformly specified for the entire network. This could be investigated in future studies. It is important to highlight that **more than 83% of the BB events** that prevailed in the case study dataset **were**

Table 5: Summary Statistics on the Results using an n-weighted mean±std.dev drawn from all routes regarding the BB predictive framework. Times in Seconds.

Evaluation Metric	Summarized Results (All Routes)
MAE Offline Regression	814.91 ± 772.57
MAE Proposed Framework	24.71 ± 6.89
Accuracy/Precision	$98.06\% \pm 1.48\%$ / $54.16\% \pm 17.03\%$
Total of BB Events	10311 / $74.24\% \pm 14.72\%$
Avg. Nr. of Stops Ahead	11.31 ± 3.46

Table 6: Experimental Results regarding the deployment of the corrective actions. The **ALL** column corresponds to aggregated results (i.e. the sum of the cor. actions and the weighted average per routes using the number of trips). Times in Seconds.

	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2
Total of Cor. Actions	57	43	1225	874	833	1101	1119	927	276	273
% Bus Holding	80.70%	88.37%	79.02%	89.25%	75.51%	95.00%	86.23%	85.01%	69.57%	86.08%
% Stop Skipping	8.77%	4.65%	9.06%	5.72%	15.61%	0.64%	5.99%	8.52%	25.36%	8.79%
% None	10.53%	6.98%	11.92%	5.03%	8.88%	4.36%	7.78%	6.47%	5.07%	5.13%
Avg. Total Holding Time	93.26	104.21	109.59	114.81	108.79	119.20	115.93	109.72	95.00	107.62
% of BB Reduction	50.98%	65.00%	66.08%	77.83%	65.48%	86.89%	72.09%	71.05%	61.83%	74.90%
AIVT w/ Actions	193	187	725	631	620	515	515	470	530	488
AIVT without Actions	193	187	732	632	614	508	502	463	525	488
AWT w/ Actions	1327	1345	872	1020	715	733	730	788	1288	1350
AWT without Actions	1327	1344	903	1034	749	751	800	839	1322	1380
% of Reduction of AWT	0.00%	0.00%	3.43%	1.35%	4.54%	2.40%	8.75%	6.08%	2.57%	2.17%

	F1	F2	G1	G2	H1	H2	I1	I2	ALL
Total of Cor. Actions	856	648	2554	2782	23	29	418	230	**14268**
% Bus Holding	83.18%	87.81%	89.78%	58.05%	73.91%	90.00%	72.48%	81.30%	**81.68%**
% Stop Skipping	10.40%	5.40%	4.42%	11.29%	21.74%	3.45%	25.12%	14.35%	**10.26%**
% None	6.43%	6.79%	5.79%	30.66%	4.35%	6.56%	2.40%	4.35%	**8.06%**
Avg. Total Holding Time	112.24	116.73	117.50	116.47	105.88	113.08	108.51	111.34	**111.16**
% of BB Reduction	68.66%	77.15%	66.91%	47.48%	63.64%	65.38%	59.07%	62.72%	**67.59%**
AIVT w/ Actions	685	928	601	597	739	586	698	498	**584.39**
AIVT without Actions	678	935	599	603	735	587	699	499	**560.41**
AWT w/ Actions	915	1070	801	888	1682	1720	1369	1501	**1038.40**
AWT without Actions	924	1082	933	983	1680	1725	1404	1547	**1043.65**
% of Reduction of AWT	0.97%	1.11%	14.15%	9.66%	-0.00%	0.00%	2.49%	2.97%	**4.46%**

forecasted accurately.

Table 6 summarizes the corrective actions implementation and their impact on passengers travel times. As expected, Holding is selected in most cases over Stop Skipping (i.e. 81.68%). However, it is important to highlight that in a substantial share of BB detection no action was taken (i.e. 30.66% for route G2) which arises from the abovementioned need to set route-specific values of η (i.e. minimum headway threshold for BB). Noticeably, the applied framework did not prove to be effective for low-frequency routes (i.e. lines A and H). This is not surprising as the BB phenomenon is most prevalent on high-frequency routes and the corrective actions deployed in this study are designed to regulate headways on high-demand routes. Furthermore, note that the low value of χ (i.e. symmetric user-defined minimum threshold for the BB likelihood required to deploy a corrective action), constrained the deployment of actions to a small subset of trips (i.e. between 3% and 7% of the total number of trips - check Table 4). and therefore yielded a reduction of only 4.46% when measured globally. This reflects however a substantial

travel time savings for those cases when BB is prevalent. Moreover, the number of BB events is reduced by 67.59% (out of the 83% of the BB forecasted). It clearly demonstrates the usefulness of our framework for this particular application - which has no parallel in the literature. It is also remarkable that this is attained without inducing an increase in the global In-Vehicle Time. Further gains could be obtained by setting optimal parameters for each route. A data driven approach to such problem could utilize the information about the actions currently deployed on each route.

6.2. Technical Discussion and Key Contributions

One of the key ideas behind this paper is that a *moving world requires explanatory models that can follow such movement* . As previously referred in Section 2.2, the traditional batch learning models cannot cope with non-stationary distributions nor concept drifts, which are key characteristics of dynamic systems - such as human mobility. Recently, the idea of coping offline and online learning together revealed promising insights, e.g. Support Vector Regression (SVR) with Kalman Filters by Yu et al. [75]. Hereby, the authors aimed to elaborate a step forward on this setup by proposing a fully non-parametric learning approach which iterates directly over the model's output. A careful inspection schema is put in place in order to verify if the recent residual's distribution is drifting from its expected parameters or not. The idea is not properly novel per se - its innovation relies instead on this pratical application, which conceeds an extra value over a specific domain area. Similarly to the stochastic sequential learning process proposed by Rodrigues and Gama [59] and applied by Moreira-Matias et al. [44] to fit ARIMA model's incrementally, we also leverage on the continuous arrival of data samples (i.e. streams) to learn online. This is also aligned with the learning mechanism employed by the stochastic gradient boosting [30], where a weak learner is additevely turned into a stronger one by approximating their residual's distribution iteratively. The key difference is that we apply this gradient directly over the output of the explanatory model, in a one-pass fashion. Technically, the main advantage of this approach is that it may converge under any type of residual distribution, as opposed to the traditional Kalman Filters, which typically assume a Gaussian noise distribution.

RF was the algorithm selected as baseline offline regressor. This choice is supported by author's previous work in Mendes-Moreira et al. [42], where RF was compared against Project Pursuit Regression (PPR) [31] and SVR throughout a set of exaustive experiments and preprocessing setups for a

similar purpose (i.e. long-term round trip time prediction). The main reason to do it is their known ability to provide pratical results without any preprocessing requirements (e.g. hyperparameter tuning or feature engineering). However, our experiments uncovered high MAE on some routes. One of the possible reasons is the shifting between round-trip and link travel times hereby studied. In spite of the low quality of base learner predictions, which served as departure point, our online framework is able to converge and improve considerably its results. In the authors' opinion, these conclusions are clearly supported by the presented results, which illustrate this capacity on any route.

However, due to this unexpected behavior of RF, two additional questions show up: 1) Is it possible to fine tune RF for this application? 2) Would our learning framework be able to perform better by departing from a better base learner? Regarding the latter, assuming that both PPR and SVR could do it under the right preprocessing setup Mendes-Moreira et al. [42], the authors decided to assess their predictive power on the present dataset. We did so by using their implementations provided by the R packages [stats] and [e1071], respectively. As an additional preprocessing task, we conduted a simple hyperparameter setting and removed the DATE feature following the RReliefF analysis illustrated in TablesA.7 and 3. Similarly to the parameter tuning task described in Section 5.2, the last ten days of January were used as validation set while the remaining ones were used to train predictive models with all possible combinations of hyperparameters. The value combination which minimized the prediction's error on the validation were selected to define our parameters values.

Such local minima were obtained for each learning algorithm using the Nelder-Mead method (i.e. simplex) [54]. On SVR, we considered the following setting's possibilities: $K = \{$linear,polynomial,sigmoid,radial$\}, \epsilon = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, $C = \{2, 3, 4\}^{\{0,1,2\}}$, $d = \{2, 3, 4\}($polynomial$), \gamma = \{1, 2, 3\}^{-n}$ (non-linear). For PPR, we considered the following instead: $\varsigma \in \{2..8\}$. The hyperparameters used on each route's dataset are depicted in Table A.9.

Then, a experimental setup similar to the one used to obtain the MAE for RF detailed in Table A.10 was followed using the best parameter setting found for each route. The results are detailed in Table A.11. These results confirm that our predictive framework would present far better results for this partcular task. However, they also uncover that some variance reduction (i.e. reduced error) can be achieved for some routes. Some of the unexplained variance on the target variable may result from the hyperparameter tunning

that both SVR and PPR benefited. Obviously, it is difficult to assess the significance level of this conclusion without a proper test, e.g. Friedman Rank Test [36]. Hence, it opens the bridge for further exploitations of such offline model/hyperparameter setting on future research on this topic, as well as on the benefits that a better departure point can bring to our predictive framework.

6.3. Learning under Drift

Despite its high applicability to the present use case in Public Transport, this framework may be applicable in a more generical way to any problem where it is necessary to predict for a rolling horizon and still deal with drifts on the underlying distribution.

From a high-level perspective, our algorithm operates in two stages: firstly, the residuals distribution produced by a given predictor is monitored by a continuous inspection schema of interest for drift detection purposes. This step aims to assess if the assumptions used to learn it (e.g. stationarity) are being violated. Secondly, a residual-based version of the parameter's inverse gradient is used to update the model whenever possible and/or directly its output. The second stage is only performed whenever an alarm is triggered on the first one, thus activating these updates.

Formally, the regression task consists on approximating the original function $f(x_i)$ using some data points part of a given *training set* (i.e. AVL data in our example). The approximation function $\hat{f}(x_i, \varepsilon)$ is **learned** to map a particular instance of the feature space X in the target one Y using a particular optimization procedure of interest (e.g. Simulated Annealing) to find a good *local minima* with respect to a given loss function (e.g. L_2). Then, our approach consists basically in start monitoring drift continuousily. Once it is detected, we start updating our model's predictions on a proportional way to the most recent prediction' residual (i.e. the *inverse gradient*), learning on a sample fashion. This procedure is depicted in Algorithm 1.

The main difference from our schema to the present one is that we assume that the drift is *always* happening somehow (i.e. incremental drift [34]). Notoriously, this schema shares some similarities with the Stochastic Gradient Descent. The two main nuances are the following: (i) the base model is always fitted using the same batch size (i.e. training set size) - instead of changing it dynamically accordingly to any convergence criteria - and; (ii) we can use different types of base learners and/or functional forms for our

35

Input: $\hat{f}$ - approximation function, ε - parameter set, Γ - monitoring window size, ω_0 - initial learning rate, ω - constant reactivability rate, ϱ - activation period, A - drift detection algorithm; X, Y-dataset

Output: $\hat{y}$ - corrected predicted outputs

$W \leftarrow \emptyset$ and $\omega_2 \leftarrow \omega_1 \leftarrow \omega_0$;

foreach $i \leftarrow 3..t$ **do**

  if *(A(W, f) == 1)* **then**

    | $\omega_i \leftarrow \omega_{i-1}$;

  else

    | $\omega_i \leftarrow 0$;

  end

  if *($|f(x_{i-1}) - \hat{f}_{i-1}| \geq |f(x_{i-2}) - \hat{f}_{i-2}| \wedge A(W, f) == 1$)* **then**

    | $\omega_i \leftarrow \omega_{i-1}(1 + \omega i - 1) : \omega_i \leq \omega_{\max}$; //increase the learning rate

  else

    | $\omega_i \leftarrow \omega_{i-1}(1 - \omega i - 1) : \omega_i \geq \omega_{\min}$; //decrease the learning rate

  end

  $\hat{y}_i \leftarrow \hat{f}(x_i, \varepsilon_i) - \omega_i(f(x_{i-1}) - \hat{f}_{i-1})$; // correct our prediction output

  $W \leftarrow W \cup (f(x_{i-1}) - \hat{f}_{i-1})$; // add elements to the head of W

  if *($|W| == \Gamma$)* **then**

    | drop an element from the tail of W;

  end

  if *($\sum_{j=i-\varrho}^{i} A(W, f) == 0$)* **then**

    | $\omega_i \leftarrow 0$; // deactivate the prediction corrections

  end

end

Algorithm 1: Pseudocode for our generic *drift-free* regression algorithm.

dependences (e.g. in this paper, we depart from a model produced by decision trees). Nevertheless, a similar version of this procedure already obtained good results on other applications (see, for instance, the incremental computation of the ARIMA weights in [44] which leverages on an offline computed model as initialization for the model's weights).

This method is applicable to any supervised learning task to be conducted over sequential data where the concept drift poses a main issue regarding the application of traditional Machine Learning techniques.

7. Conclusions

A novel real-time framework to prevent BB from occurring is proposed in this paper. It combines historical and real-time AVL data to predict the occurrence of BB events at downstream stops. The prediction output is then used to select and deploy automatic corrective actions. This framework consists of advanced ML methods which are able to gain foresight on the BB process. Experiments were conducted using a large-scale dataset of real world data collected in Porto, Portugal. The application yielded a reduction of 68% in the number of BB events. The results demonstrate that this framework can be readily deployed for mass transit systems across the world. Moreover, it is estimated that it could have a real impact on passengers experience by decreasing the average expected waiting time by approximately 5% without increasing in-vehicle times.

Future work includes carrying out experiments to optimally set the parameters η (i.e. an headway-based minimum threshold to consider a BB event) and χ (i.e. a minimum BB likelihood threshold to deploy a corrective action) for each individual route and possibly also time-dependent. Further research can use data about the corrective actions deployed on each route for this purposes. Moreover, the assumptions about the functional form of the headway distribution (i.e. Gaussian) and their parameter calculus may be too simplistic in some cases. Further research may consider handling various types of distributions. The parameter estimation can also be improved by employing change detection techniques (e.g. the CUmulative SUM algorithm [53]) that are able to avoid the inclusion of outliers in the calculus of the distribution's parameter - instead of using a simple median of the recent residuals). Experiments on improving the predictive power of the base offline regressor can also be futurely explored to provide novel insights for this particular application. Three concrete suggestions in this research line include (1) testing other types of learners, e.g. SVR; (2) feature engineering processes and; (3) study custom loss-functions for this particular problem (i.e. weight peak-hour residuals to increase precision/recall of BB event prediction).

The framework presented in this paper could ultimately be embedded into a decision support system that can be deployed in control rooms of PT agencies and operators.

8. Acknowledgements

References

[1] Abkowitz, M., Engelstein, I., 1984. Methods for maintaining transit service regularity. Tech. rep.

[2] Abkowitz, M., Tozzi, J., 1987. Research contributions to managing transit service reliability. Journal of advanced transportation 21 (1), 47–65.

[3] Bin, Y., Zhongzhen, Y., Baozhen, Y., 2006. Bus arrival time prediction using support vector machines. Journal of Intelligent Transportation Systems 10 (4), 151–158.

[4] Bishop, C. M., et al., 1995. Neural networks for pattern recognition.

[5] Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

[6] Cappé, O., Godsill, S., Moulines, E., 2007. An overview of existing methods and recent advances in sequential monte carlo. Proceedings of the IEEE 95 (5), 899–924.

[7] Carnaghi, P., 2014. Decision support systems in transportation: the case of surface urban public transport.

[8] Carrascal, U., 2012. A review of travel time estimation and forecasting for advanced traveler information systems. Master's thesis, Universidad del Pais Vasco.

[9] Cathey, F., Dailey, D., 2003. A prescription for transit arrival/departure prediction using automatic vehicle location data. Transportation Research Part C: Emerging Technologies 11 (3), 241–264.

[10] Cats, O., 2014. Regularity-driven bus operation: Principles, implementation and business models. Transport Policy 36, 223–230.

[11] Cats, O., Burghout, W., Toledo, T., Koutsopoulos, H. N., 2010. Evaluation of real-time holding strategies for improved bus service reliability. In: 13th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 718–723.

[12] Cats, O., Larijani, A. N., Koutsopoulos, H. N., Burghout, W., 2011. Impacts of holding control strategies on transit performance. Transportation Research Record: Journal of the Transportation Research Board 2216 (1), 51–58.

[13] Cats, O., Larijani, A. N., Ólafsdóttir, Á., Burghout, W., Andréasson, I. J., Koutsopoulos, H. N., 2012. Bus-holding control strategies. Transportation Research Record: Journal of the Transportation Research Board 2274 (1), 100–108.

[14] Cats, O., Loutos, G., 2013. Real-time bus arrival information system–an empirical evaluation. In: Proceedings of the 16th International IEEE conference on Intelligent Transport Systems (ITSC).

[15] Cats, O., Rufi, F. M., Koutsopoulos, H., 2014. Optimizing the number and location of time point stops. Public Transport 6 (3), 215–235.

[16] Ceder, A., 2007. Public transit planning and operation: theory, modeling and practice. Elsevier, Butterworth-Heinemann.

[17] Chapman, R., Michel, J., 1978. Modelling the tendency of buses to form pairs. Transportation Science 12 (2), 165–175.

[18] Chen, M., Liu, X., Xia, J., Chien, S., 2004. A dynamic bus-arrival time prediction model based on apc data. Computer-Aided Civil and Infrastructure Engineering 19 (5), 364–376.

[19] Chien, S., Ding, Y., Wei, C., 2002. Dynamic bus arrival time prediction with artificial neural networks. Journal of Transportation Engineering 128 (5), 429–438.

[20] Daganzo, C., 2009. A headway-based approach to eliminate bus bunching. Transportation Research Part B 43 (10), 913–921.

[21] Daganzo, C., Pilachowski, J., 2011. Reducing bunching with bus-to-bus cooperation. Transportation Research Part B: Methodological 45 (1), 267–277.

[22] Dawid, A., 1984. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. Journal of the Royal Statistical Society. Series A (General) 147, 278–292.

[23] Delgado, F., Munoz, J. C., Giesen, R., 2012. How much can holding and/or limiting boarding improve transit performance? Transportation Research Part B: Methodological 46 (9), 1202–1217.

[24] Delgado, F., Muñoz, J. C., Giesen, R., Cipriano, A., 2009. Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. Transportation Research Record: Journal of the Transportation Research Board 2090 (1), 59–67.

[25] Dong, J., Zou, L., Zhang, Y., 2013. Mixed model for prediction of bus arrival times. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2918–2923.

[26] Eberlein, X., Wilson, N., Bernstein, 2001. The holding problem with real-time information available. Transportation Science 35 (1), 1–18.

[27] El-Geneidy, A., Horning, J., Krizek, K., 2011. Analyzing transit service reliability using detailed data from automatic vehicular locator systems. Journal of Advanced Transportation 45 (1), 66–79.

[28] Feuerstein, S., Pribyl, B., 1997. Oracle PL/SQL Programming (2Nd Ed.). O'Reilly & Associates, Inc., Sebastopol, CA, USA.

[29] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F., 2015. Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp. 2944–2952.

[30] Friedman, J., 2002. Stochastic gradient boosting. Computational Statistics & Data Analysis 38 (4), 367–378.

[31] Friedman, J., Stuetzle, W., 1981. Projection pursuit regression. Journal of the American statistical Association 76 (376), 817–823.

[32] Fu, L., Yang, X., 2002. Design and implementation of bus-holding control strategies with real-time information. Transportation Research Record: Journal of the Transportation Research Board 1791 (1), 6–12.

[33] Gama, J., 2010. Knowledge discovery from data streams. Chapman and Hall/CRC.

[34] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A., 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46 (4), 44:1–44:37.

[35] Gentile, G., Nguyen, S., Pallottino, S., 2005. Route choice on transit networks with online information at stops. Transportation science 39 (3), 289–297.

[36] Iman, R., Davenport, J., 1980. Approximations of the critical region of the friedman statistic. Communications in Statistics-Theory and Methods 9 (6), 571–595.

[37] Jeong, R., Rilett, L., 2005. Prediction model of bus arrival time for real-time applications. Transportation Research Record: Journal of the Transportation Research Board 1927 (1), 195–204.

[38] Kieu, L., Bhaskar, A., Chung, E., September 2012. Benefits and issues for bus travel time estimation and prediction. In: Australasian Transport Research Forum 2012. Perth, WA.

[39] Koutsopoulos, H. N., Wang, Z., 2007. Simulation of urban rail operations: Application framework. Transportation Research Record: Journal of the Transportation Research Board 2006 (1), 84–91.

[40] Mazloumi, E., Rose, G., Currie, G., Moridpour, S., 2011. Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction. Engineering Applications of Artificial Intelligence 24 (3), 534–542.

[41] McClelland, J. L., Rumelhart, D. E., Group, P. R., et al., 1986. Parallel distributed processing. Explorations in the microstructure of cognition 2.

[42] Mendes-Moreira, J., Jorge, A., de Sousa, J., Soares, C., 2012. Comparing state-of-the-art regression methods for long term travel time prediction. Intelligent Data Analysis 16 (3), 427–449.

[43] Moreira-Matias, L., Ferreira, C., Gama, J., Mendes-Moreira, J., de Sousa, J., 2012. Bus bunching detection by mining sequences of headway deviations. In: Advances in Data Mining. Applications and Theoretical Aspects. Vol. 7377 of LNCS. Springer., pp. 77–91.

[44] Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L., 2013. On predicting the taxi-passenger demand: A real-time approach. In: Progress in Artificial Intelligence. Vol. 8154 of LNCS. Springer, pp. 54–65.

[45] Moreira-Matias, L., Gama, J., Mendes-Moreira, J., Freire de Sousa, J., 2014. An incremental probabilistic model to predict bus bunching in real-time. In: Advances in Intelligent Data Analysis XIII. Vol. 8819 of LNCS. Springer International Publishing, pp. 227–238.

[46] Moreira-Matias, L., Mendes-Moreira, J., Freire De Sousa, J., Gama, J., 2015. Improving mass transit operations by using avl-based systems: A survey. Intelligent Transportation Systems, IEEE Transactions on 16 (4), 1636–1653.

[47] Munizaga, M. A., Palma, C., 2012. Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from santiago, chile. Transportation Research Part C: Emerging Technologies 24, 9–18.

[48] Muñoz, J. C., Cortés, C. E., Giesen, R., Sáez, D., Delgado, F., Valencia, F., Cipriano, A., 2013. Comparison of dynamic control strategies for transit operations. Transportation Research Part C: Emerging Technologies 28, 101–113.

[49] Neapolitan, R. E., 2012. Probabilistic reasoning in expert systems: theory and algorithms. CreateSpace Independent Publishing Platform.

[50] Newell, G., Potts, R., 1964. Maintaining a bus schedule. In: 2nd Australian Road Research Board. Vol. 2. pp. 388–393.

[51] Nunes, R., Moreira-Matias, L., Ferreira, M., oct. 2012. Using exit time predictions to optimize self automated parking lots. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 302 –307.

[52] Oracle, 2016. Oracle database documdocument library. Acessed January 2016.
URL http://www.oracle.com/pls/db102/homepage

[53] Page, E., 1954. Continuous inspection schemes. Biometrika 41 (1/2), 100–115.

[54] Powell, M., 1973. On search directions for minimization algorithms. Mathematical Programming 4 (1), 193–201.

[55] Powell, W., Sheffi, Y., 1983. A probabilistic model of bus route performance. Transportation Science 17 (4), 376–404.

[56] R Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org

[57] Rajbhandari, R., 2005. Bus arrival time prediction using stochastic time series and markov chains. Ph.D. thesis, New Jersey Institute of Technology.

[58] Robnik-Šikonja, M., Kononenko, I., 1997. An adaptation of relief for attribute estimation in regression. In: Machine Learning: Proceedings of the Fourteenth International Conference (ICML 97). pp. 296–304.

[59] Rodrigues, P., Gama, J., 2009. A system for analysis and prediction of electricity-load streams. Intelligent Data Analysis 13 (3), 477–496.

[60] Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65 (6), 386.

[61] Schaul, T., Zhang, S., Lecun, Y., 2013. No more pesky learning rates. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). Vol. 28. JMLR Workshop and Conference Proceedings, pp. 343–351.

[62] STCP - Sociedade de Transportes Colectivos do Porto, October 2013. URL http://www.stcp.pt/

[63] Stone, G. O., 1986. An analysis of the delta rule and the learning of statistical associations. Parallel distributed processing: Explorations in the microstructure of cognition 1, 444–459.

[64] Strathman, J., 1998. Automated bus dispatching, operations control and service reliability: analysis of tri-met baseline service date. Tech. rep., University of Washington.

[65] Strathman, J., Kimpel, T., Dueker, K., 1999. Automated bus dispatching, operations control and service reliability. Transportation Research Record 1666, 28–36.

[66] Strathman, J., Kimpel, T., Dueker, K., Transportation Northwest, 2000. Bus transit operations control: review and an experiment involving tri-met's automated bus dispatching system. Tech. rep., Transportation Northwest, Department of Civil Engineering, University of Washington.

[67] Strathman, J. G., Hopper, J. R., 1993. Empirical analysis of bus transit on-time performance. Transportation Research Part A: Policy and Practice 27 (2), 93–100.

[68] Sun, A., Hickman, M., 2008. The holding problem at multiple holding stations. In: Computer-aided systems in public transport. Springer, pp. 339–359.

[69] TCRP, 2003. Transit Capacity and Quality of Service Manual. Vol. 100. Transportation Research Board.

[70] Toledo, T., Cats, O., Burghout, W., Koutsopoulos, H., 2010. Meso-scopic simulation for transit operations. Transportation Research Part C: Emerging Technologies 18 (6), 896 – 908.

[71] Turnquist, M., Blume, S., 1980. Evaluating potential effectiveness of headway control strategies for transit systems. Transportation Research Record (746).

[72] van Oort, N., 2011. Service reliability and urban public transport design. Netherlands TRAIL Research School.

[73] Wardman, M., 2004. Public transport values of time. Transport policy 11 (4), 363–377.

[74] Widmer, G., Kubat, M., 1996. Learning in the presence of concept drift and hidden contexts. Machine Learning 23 (1), 69–101.
URL http://dx.doi.org/10.1023/A%3A1018046501280

[75] Yu, B., Yang, Z., Chen, K., Yu, B., 2010. Hybrid model for prediction of bus arrival times at next station. Journal of Advanced Transportation 44 (3), 193–204.

A. Annexures

A.1. Case Study

A.1.1. Details About the Studied Routes

Line A is a commuter line between downtown and *Vila D'este*, a large poor neighbourhood located in the southern edge of the Douro river which trespasses many rural areas. Line B is a major urban line that connects the major city center market street to a luxurious neighbourhood on the city seaside (*Castelo do Queijo*). Line C is also a major urban line between *Viso* (an important neighbourhood in Porto) and *Sá da Bandeira*, a downtown bus hub. Line D connects downtown to *Hospital São João* (HSJ), an important bus/light train terminal in the northern part of the city. Line E connects downtown to a highly populated neighbourhood in the east (*S.Roque*). Line F is an arterial urban line. It traverses the main points of interest in the city by connecting two important market streets: *Bolhão* - located in downtown - and *Mercado da Foz*, located in the most luxurious neighbourhood in the city. Line G connects the city downtown to the farthest large-scale neighbourhood in the urban agglomeration (*Maia*). Line H departs from an important terminal located on the city outskirts (*Marquês*) to a highly dense residential area to the east (*Rio Tinto*). Finally, Line I connects the two major transport hubs in the city: *Trindade*, that joins all the five light tram lines in the city and *Sto. Ovídeo*, which is the southernmost light tram station that provides bus connections to most of the neighbourhoods located in this region.

Eight of these routes are depicted on the road network in the urban area of Porto in Fig. A.5. The orange dots represent the bus stops of each route.
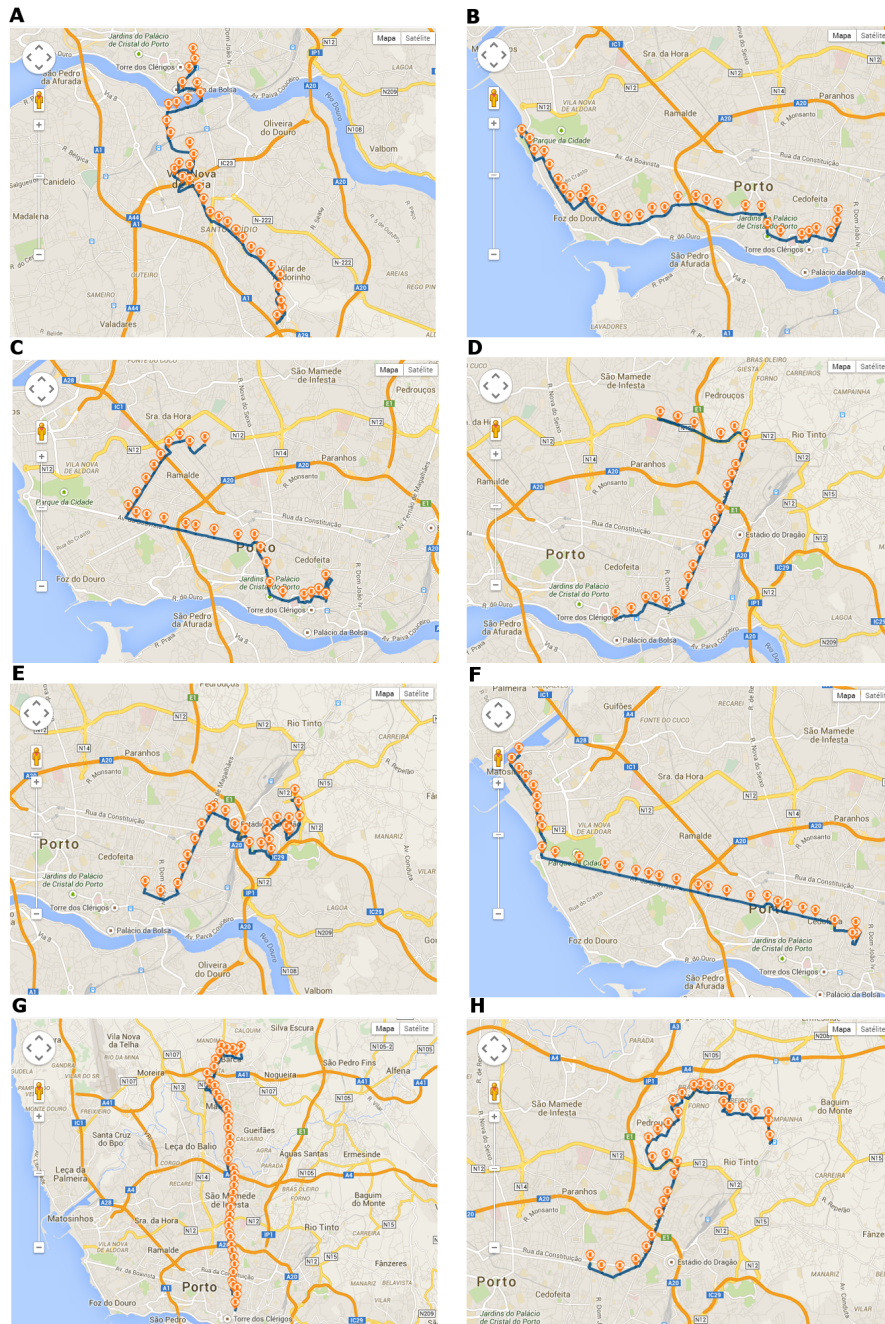
Figure A.5: Illustration of some routes (one per line) considered over a geographical representation of the road network in Porto, Portugal. Image obtained from [62].

A.1.2. Exaustive Data Statistics

Table A.7: `RReliefF` results detailed per route.

ROUTE	SHIFT	TRIP	HOLIDAY	TIMESTAMP	DATE	WEEKDAY	START_ST	END_ST
A1	-0.013	0.595	0.000	0.228	-0.104	-0.043	-0.030	-0.010
A2	-0.017	-0.070	-0.007	-0.013	-0.393	-0.067	0.000	0.000
B1	-0.060	-0.037	0.000	-0.008	-0.377	-0.113	0.000	0.000
B2	-0.093	-0.017	-0.010	-0.019	-0.388	-0.100	0.000	0.000
C1	-0.037	-0.017	0.000	-0.019	-0.428	-0.063	0.000	0.000
C2	-0.057	-0.017	0.000	-0.011	-0.365	-0.110	-0.003	0.000
D1	-0.020	-0.033	0.000	-0.008	-0.404	-0.093	0.000	0.000
D2	-0.043	-0.017	0.000	-0.002	-0.402	-0.070	0.000	0.000
E1	-0.050	-0.037	-0.017	-0.030	-0.387	-0.027	0.000	0.000
E2	-0.063	-0.027	0.000	-0.011	-0.391	-0.053	0.000	0.000
F1	-0.013	-0.027	0.000	-0.018	-0.414	-0.037	0.000	0.000
F2	-0.027	-0.003	0.000	-0.012	-0.445	-0.017	0.000	0.000
G1	-0.073	-0.020	-0.013	-0.003	-0.358	-0.083	-0.010	-0.013
G2	-0.117	-0.010	-0.007	-0.034	-0.365	-0.080	0.000	0.000
H1	-0.020	-0.023	-0.023	-0.017	-0.385	-0.037	0.000	0.000
H2	-0.023	-0.023	-0.020	-0.005	-0.397	-0.043	0.000	0.000
I1	-0.040	-0.027	-0.020	-0.016	-0.409	-0.073	-0.010	0.000
I2	-0.057	-0.020	-0.023	-0.011	-0.390	-0.087	0.000	0.000

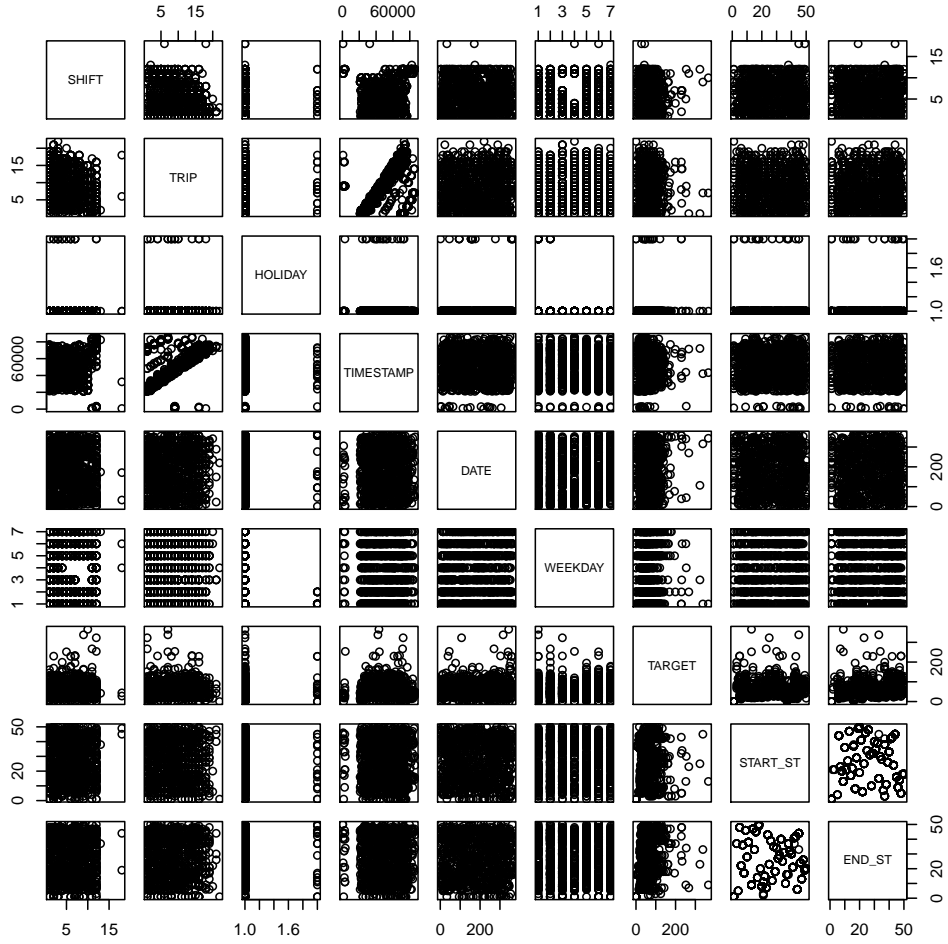Figure A.6: Pairwise Scatterplot of the available variables to study correlations among the input data for a subsample of G1.

(a) Route A1.

(b) Route B1.

(c) Route C1.

(d) Route D2.

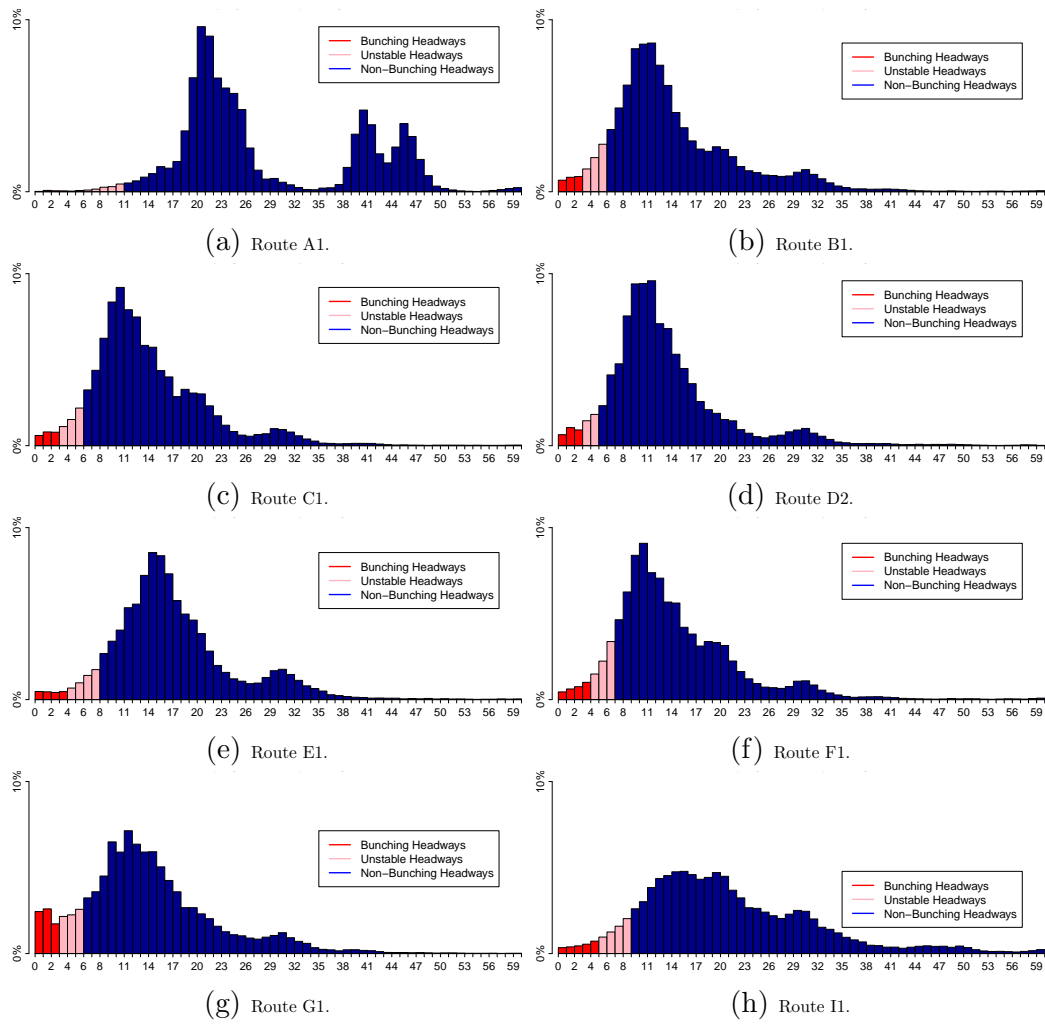(e) Route E1.

(f) Route F1.

(g) Route G1.

(h) Route I1.

Figure A.7: Sample-based Headway (discrete) Distribution for eight routes of this study during the peak hours(truncated between 0 and 1h). Times are in minutes.

A.2. Detailed Experimental Setup

A.2.1. Parameter/Hyperparameter Tuning

Table A.8: Resulting Parameter Setting.

Route	ϕ	ω_{min}	ω_{max}	β^2	τ	Route	ϕ	ω_{min}	ω_{max}	β^2	τ
A1	0.100	0.010	0.300	0.010	10	**A2**	0.100	0.010	0.300	0.010	10
B1	0.050	0.010	0.400	0.300	5	**B2**	0.050	0.010	0.400	0.300	3
C1	0.050	0.005	0.300	0.300	5	**C2**	0.050	0.010	0.400	0.300	5
D1	0.075	0.010	0.400	0.500	3	**D2**	0.100	0.010	0.400	0.300	3
E1	0.050	0.005	0.300	0.300	5	**E2**	0.100	0.005	0.300	0.100	10
F1	0.050	0.005	0.300	0.300	5	**F2**	0.050	0.005	0.300	0.300	5
G1	0.050	0.005	0.300	0.300	5	**G2**	0.050	0.005	0.300	0.300	5
H1	0.050	0.005	0.300	0.100	5	**H2**	0.100	0.005	0.300	0.100	5
I1	0.050	0.005	0.300	0.300	5	**I2**	0.050	0.005	0.300	0.300	5

Table A.9: Resulting Hyperparameter Setting.

Route	Kernel	γ	C	ϵ	ς	Route	Kernel	γ	C	ϵ	ς
A1	linear	NA	1	0.2	2..4	**E1**	linear	NA	9	0	2..6
A2	linear	NA	9	0.2	2..4	**E2**	linear	NA	4	0	2..4
B1	linear	NA	4	0	2..4	**F1**	linear	NA	9	0	3..6
B2	linear	NA	4	0	2..4	**F2**	linear	NA	16	0	2..5
C1	linear	NA	9	0	2..4	**G1**	sigmoid	3^{-n}	9	0	2..5
C2	linear	NA	16	0	2..4	**G2**	sigmoid	2^{-n}	16	0	4..6
D1	linear	NA	16	0	2..4	**H1**	sigmoid	3^{-n}	16	0	3..5
D2	linear	NA	16	0	2..5	**H2**	linear	NA	9	0	2..4
I1	linear	NA	16	0.2	2..4	**I2**	linear	NA	16	0	2..4

A.2.2. Evaluation Metrics

It is possible to divide the evaluation of our framework into three distinct dimensions: (i) the mean absolute error (MAE) of the headway's predictions, (ii) the BB detection accuracy and (iii) the effect of the actions deployed in the PT network on travellers.

Concerning the first dimension, (i) a prequential evaluation [22] was performed by evaluating just the prediction made for the LTT prediction performed for the next bus stop. Consequently, the MAE for a given route is

computed as follows

$$MAE = \frac{1}{n \times s} \sum\nolimits_{k=1}^{n} \sum\nolimits_{i=1}^{s} H'R_k^i \qquad (A.1)$$

In (ii) the Accuracy, the Precision and the Recall were employed to evaluate the BB event detection framework. A Weighted Accuracy was also employed to give greater weight to a false negative versus a false positive, in the detection of BB events. It can be computed as

$$WAcc = \frac{(10 \times TP) + TN}{(10 \times TP) + TN + FP + (10 \times FN)} \qquad (A.2)$$

The Average Number of Stops Ahead is also displayed to show the forecasting horizon that this framework can yield.

The most direct metric for dimension (iii) concerns the percentage of BB reduced by employing our automatic control actions. Such ratio can be computed as follows

$$BB_{reduction} = \frac{\texttt{BB\_Trips\_Without\_Actions} - \texttt{BB\_Trips\_With\_Actions}}{\texttt{BB\_Trips\_Without\_Actions}}$$
$$(A.3)$$

Nevertheless, the ultimate motivation to avoid the occurrence of BB events is to improve the global quality of the service provisioned. A BB event does decrease the passengers' perception of the service quality. Moreover, it also yields longer Passenger Waiting Times for passengers waiting at downstream stops. However, the deployment of corrective actions can potentially increase Passenger In-Vehicle Time due to prolonged times at stops and on-board inflicted by holding or stop skipping [11]. It is therefore necessary to consider both the **Average In-Vehicle Time** (AIVT) and the **Average Waiting Time** (AWT) (already defined in eqs. (3,6) when evaluating alternative operational plans. By avoiding the occurrence of BB events, it is expected to reverse the well-known snowball effect of the BB process and hence reduce global AWT on a given route. The deployment should ensure that this is achieved without compromising global AIVT. In order to evaluate the success of such minimization task, two large set of simulations were performed: (SIM1) no actions were deployed on the route and (SIM2) actions were deployed accordingly to the framework described in Section 4.6.

In SIM2, two additional ghost trips were introduced whenever a BB alarm is triggered. This is done by re-running the two affected trips from the

beginning applying the necessary corrective actions and not applying any action at all on the ghost trips. The idea is to estimate the variation on the dwell times experienced by both trips to predict the real impact on the vehicle's LTT and, consequently, on the AIVT and AWT. Such ghost trips also serve to evaluate the accuracy of our corrective actions in preventing the occurrence of such BB event (i.e. act_{ACC}).

A.2.3. Demand Profile Generation Procedure

Let o_{max} be the maximum passenger capacity of a bus running a given route. The occupancy of a given vehicle k after departing from a given bus stop i is given as follows

$$o_k^i = o_k^{i-1} + bo_k^i - al_k^i : o_k^i \leq o_{max} \tag{A.4}$$

where bo_k^i and al_k^i denote respectively the number of boarding and alighting passengers for trip k at bus stop i. In the absence of empirical data to calculate such values, we assumed that, on high frequent routes, the passengers arrival process follows an inhomogeneous Poisson process with a given rate $\lambda(t)$.

It is known that the timetables are designed to accommodate variations in passenger demand levels by setting different service frequencies for different time periods and routes (chapter 4 in [16]). In order to introduce some perturbations in passenger demand, we sampled values from (λ) using a **Gaussian** p.d.f.. Such probability distribution is defined based on the frequency $f_{k,k+1}$ and some user-defined parameter $0 < \upsilon << 1$ which basically sets the amount of *white noise* introduced on our demand generation model. The sampling process is defined as follows

$$\lambda(k) \sim \mathcal{N}\left(\mu = \upsilon \times f_{k,k+1}, \sigma = \upsilon^3 \times f_{k,k+1}\right) : \lambda_{min} \leq \lambda(k) \leq \lambda_{max}, \lambda(k) \in \mathbb{N} \tag{A.5}$$

where $\lambda_{min}, \lambda_{max}$ are user defined boundaries for $\lambda(k)$. υ denotes the percentage of the frequency to be used when calculating passenger arrivals and $\lambda_{min}, \lambda_{max}$ are a minimum/maximum threshold for the value of $\lambda(k)$. Based on such *p.d.f.* definition, values for $\lambda(k)$ can be *sampled* for each trip k.

From empirical evidence, it is also known that passenger demand also varies along the route (e.g. [47]). This is captured by incorporating a linear descendent demand factor for each bus stop b_i, i.e. df_i. It can be computed as

$$df_i = \begin{cases} \frac{2 \times (s-i+1)}{s} & \textbf{if } i \leq s, \\ 0 & \text{otherwise.} \end{cases} \tag{A.6}$$

Based on eqs. (A.5,A.6), it is possible to infer the calculus of bo_k^i as follows

$$de_{k-1,k}^i = \left(H_{k-1,k}^i \times \lambda(k) \times df_i \right) \tag{A.7}$$

$$bo_k^i = de_{k-1,k}^i + Nbo_{k-1}^i : bo_k^i \in \mathbb{N} \tag{A.8}$$

where Nbo_{k-1}^i stands for the number of passengers that were not allowed to board on the vehicle $k-1$ and $de_{k-1,k}^i$ is the number of passengers arrived to the stop b_i during the headway between k and $k-1$ (demand generated during the period of $H_{k-1,k}^i$). $Nbo_{k-1}^i > 0$ whenever the vehicle $k-1$ rides *full* after traversing stop i or if the stop i is *skipped* by bus k as a consequence of a corrective action.

It is assumed that passengers trip length varies between 25% and 50% of the respective bus route. The user-defined parameter $25\% \leq \varphi < 50\%$ is introduced where the number of stops traversed by a given passenger z to the trip k, i.e. $ns_{z,k}$ is assumed to follow an **lognormal** distribution, as defined on the equation below

$$ns_{z,k} \sim \ln\mathcal{N}\left(\mu = \ln\left(\varphi \times df_{bo_{z,k}} \times s\right), \sigma = \ln\left(\varphi^3 \times df_{bo_{z,k}} \times s\right)\right),$$
$$\text{subject to: } ns_{z,k} \geq 1, \mu > \sigma > 1, ns_{z,k} \in \mathbb{N} \tag{A.9}$$

where $bo_{z,k}$ represents the boarding stop of the passenger z on the trip k. Consequently, it is possible to obtain the alighting stop of z in k as

$$as_{z,k} = bo_{z,k} + ns_{z,k} : as_{z,k} < s \tag{A.10}$$

Like the boardings, the alightings also assume some realistic stochasticity by being sampled from a predefined distribution (and not from an exact mathematical definition). Again, the demand factor df_i is employed (i.e. the passengers boarded on the beginning of the routes are *likely* to traverse more stops than the ones boarded on its end). Let $fas(z, i, k)$ be a *boolean* function defined as follows

$$fas(z, i, k) = \begin{cases} 1 & \textbf{if } as_{z,k} = i, \\ 0 & \text{otherwise.} \end{cases} \tag{A.11}$$

The number of alightings of a vehicle/trip k at bus stop b_j can be hence computed as follows

$$al_k^i = \sum_{z=1}^{B^i} \sum_{j=1}^{i-1} fas(z, j, k) : al_k^s = o_k^{s-1} \tag{A.12}$$

Although being enough to compute $AIVT$, the definitions in eqs. (A.8,A.12) do not allow to compute the AWT as the arrival time of a given passenger z to a bus stop b_j, $PAV_{z,k}^{b_j}$ is unknown. To obtain such values, we reverse the effects imposed by the abovementioned assumption by inferring passengers arrival times, $PAV_{z,k}^{b_j}$, from the respective exponential distribution Exp $(\lambda(k))$. This is performed using the following steps: (a) $de_{k-1,k}^i + 1, \forall i, k$ values are sampled from Exp $(\lambda(k))$ to express the time between each passenger arrival; (b) the values are normalized to let their sum meet the total elapsed time $H_{k-1,k})$ by dividing each sampled value by their total sum and then multiplying all of them by $H_{k-1,k}$; (c) the arrival times are then incrementally summed to express the time elapsed from the departure of b_j to each passenger arrival time $PAV_{z,k}^{b_j}$ - which will force one of these values to be the total sum of values (i.e. $H_{k-1,k}$), and (d) the latter value is then removed to obtain the set of $PAV_{z,k}^{b_j}$ for the demand generated on b_j between the departures of $k-1$ and k, $de_{(}k-1,k)^i$.

The consequences of deploying a given corrective action on a given trip k also have to be captured in the simulation model by representing its effects on LTT. Let $\Delta_{BH,k}^j$ and $\Delta_{SS,k}^j$ stand for the change in arrival time of trip k to the following bus stops provoked by deploying a Bus Holding or Stop Skipping action at bus stop b_j, respectively. Such changes can be computed as follows

$$\mathbb{T}_k^g = T_k^g + \Delta_{BH,g}^j + \Delta_{SS,g} \tag{A.13}$$

$$\Delta_{BH,k}^g = \sum_{i=j+1}^{g-1} HT_g^i : g > j \tag{A.14}$$

$$\Delta_{SS,k}^g = -dwT_k^j \tag{A.15}$$

$$dwT_k^j = de_{k-1,k} \times \xi + dwT_{min} : dwT_k^j \leq dwT_{max} \tag{A.16}$$

where dwT_{min}, dwT_{max} are two user-defined boundaries for the dwell time and $\xi > 0$ is a user-defined constant boarding time per passenger (which will correspond to an excess/reduction). Consequently, $\mathbb{T}_k^g$ denotes the LTT of k affected by the deployment of a corrective action on the network. However, by influencing headway stability, the characteristic recursive effect of the BB process (described in the introductory Section) is **reversed** (e.g. if some holding is imposed on k, bus $k+1$ will experience shorter dwell times as *some* of the demand will be accommodated by k). Such effects are also accounted on the simulation SIM2 by devising the following first order relationships

$$\Delta_{BH,k+1}^g = T_{k+1}^g - (\Delta bo_k^g \times \xi) + \Delta_{BH,k+1}^{g-1} \tag{A.17}$$
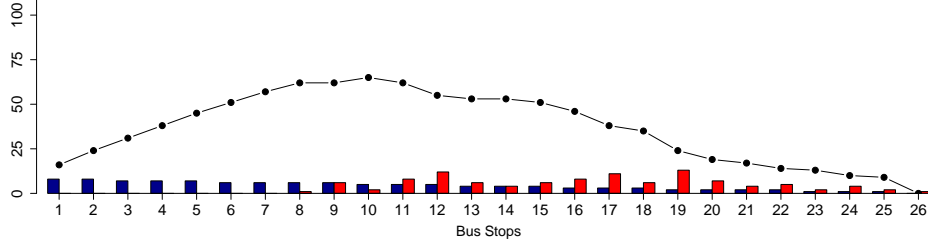
Figure A.8: Demand Profile Generated for a given trip on the route C1. The red/blue bars represent the alightings/boardings on each stop, while the dashed line represents the bus occupancy's evolution throughout the trip.

$$\Delta^g_{SS,k+1} = T^g_{k+1} - (\Delta bo^g_k \times \xi) + \Delta^{g-1}_{SS,k+1} \tag{A.18}$$

where Δbo^g_k stands by the change in the number of boarding passengers for trip k at bus stop b_g attributed to the corrective action deployment. Note that the recursive relationship imposed by eqs. (A.17,A.18]) is not necessarily constrained to the trip subsequent to the corrective action deployed $(k, k+1)$, but also to the subsequent ones $(> k + 1)$ in a **snow ball effect**.

The parameters of this demand profile generation procedure were set to the following values: $\xi = 3$, $dwT_{min} = 10$, $dwT_{max} = 90$, $\upsilon = 0.2$, $\lambda_{min} = 60$, $\lambda_{max} = 180$, $\varphi = 0.25$ (all times in seconds). An illustrative example of the demand profile generated by this model is illustrated in Fig. A.8. The results of the experimental simulations described above are presented in the following section.

A.3. Additional Results and Discussion

A.3.1. Exaustive Results

Table A.10: Experimental Results regarding the BB predictive framework. The **ALL** column corresponds for aggregated results (averaged for the prediction errors and summed for the total numbers of BB events). Times in Seconds.

	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2
MAE offline regression RF	63.79	83.77	1671.54	765.33	1356.96	643.39	277.78	174.58	255.39	363.91
MAE inter-trip update	31.08	33.87	114.16	62.17	97.87	92.91	49.26	39.31	33.65	91.56
MAE incremental update	30.38	27.97	26.47	17.67	24.14	26.35	34.78	27.74	24.78	16.30
Accuracy	99.34%	99.45%	96.96%	97.86%	97.99%	96.34%	98.57%	98.24%	99.49%	99.24%
Weighted Accuracy	98.38%	97.13%	93.86%	91.81%	93.97%	93.57%	94.41%	93.06%	97.15%	97.46%
Precision	13.85%	36.73%	49.48%	65.97%	65.88%	40.85%	74.30%	72.51%	84.75%	62.09%
Recall	47.37%	41.86%	83.52%	73.61%	81.81%	83.18%	84.54%	78.08%	82.13%	81.57%
Avg. Nr. of Stops Ahead	5.42	3.44	13.18	15.99	11.85	14.78	9.02	8.91	10.21	10.67
Correct BB Predictions	9	18	613	597	558	460	853	691	239	172
Real BB Events	19	43	734	811	682	553	1009	885	291	211
	F1	**F2**	**G1**	**G2**	**H1**	**H2**	**I1**	**I2**	**ALL**	
MAE offline regression RF	1475.72	1871.01	473.61	2776.57	1719.42	241.56	290.39	157.77	**814.91**	
MAE trip-based update	124.99	148.85	40.65	123.76	105.88	34.40	39.42	31.76	**71.98**	
MAE stop-based update	22.67	13.21	31.78	27.47	19.05	12.65	22.49	38.81	**24.71**	
Accuracy	97.08%	97.83%	96.62%	93.83%	99.81%	99.76%	98.62%	98.44%	**98.06%**	
Weighted Accuracy	94.56%	95.52%	95.72%	91.50%	99.19%	99.01%	94.70%	92.23%	**95.19%**	
Precision	41.53%	45.70%	69.44%	51.67%	40.00%	42.42%	69.39%	48.33%	**54.16%**	
Recall	83.07%	83.24%	94.47%	87.96%	58.82%	60.87%	78.87%	51.56%	**74.25%**	
Avg. Nr. of Stops Ahead	13.88	15.08	12.96	14.51	11.81	6.05	13.90	11.96	**11.31**	
Correct BB Predictions	363	303	1811	1497	10	14	306	116	**8630**	
Real BB Events	437	364	1917	1702	17	23	388	225	**10311**	

Table A.11: Additional Experimental Results regarding the Offline Regression Task using SVR and PPR.

Route	SVR	PPR	Route	SVR	PPR
A1	62.75	74.49	**A2**	82.98	85.58
B1	1844.70	1585.22	**B2**	851.45	838.06
C1	1319.09	1349.62	**C2**	675.48	775.74
D1	283.30	236.08	**D2**	158.73	178.06
E1	238.38	232.46	**E2**	370.44	417.40
F1	1647.2	1547.96	**F2**	1956.71	1878.79
G1	424.00	492.14	**G2**	2837.99	1888.79
H1	1752.36	1430.42	**H2**	246.63	216.88
I1	283.27	194.69	**I2**	196.16	185.85

A.4. Potential Deployment and Impact

The main prerequisites for the application of the proposed framework in real-world operational control of most major PT companies worldwide are

already fulfilled: the existence of AVL data in real time, a control center that monitors these data also in real time and, finally, the means to establish communication between the control center and the drivers. The same cannot be said about the subjective conditions in terms of management perceptions. The prevailing attitude among PT companies is to regard BB as almost inevitable, considered to be a constant feature of bus service. Consequently, these companies do not assign the necessary means for its resolution. This is a question that has to be faced in the dialogue between researchers and companies, and the remaining of this section is a preliminary essay to argue its importance.

The value of preventing BB is not limited to operational costs and the direct impact on passengers' experience in terms of travel times and crowding, but is also related to the overall perception of service quality. The implementation of an operational system such as the one proposed in this paper has therefore implications on the overall service perception. The feasibility of implementing such a system depends on the estimated operational costs that arise in case of BB. The following is an attempt to quantify these costs.

Each time a BB event occur, the bus driver and the vehicle of the subsequent trip may experience delays because the trip takes longer than planned. This delay is, at most, the frequency of the respective route, typically a short one; otherwise the occurrence of bus bunching would be unlikely. Since the occurrence of BB is caused by delays in the front bus and advances in the successive, the extra time $ET_c = f_{c,c+1} - H_c$ spent by the front bus c is in the interval $[0, ET_c]$. TAssuming that no measures are taken in order to regulate the headway, it is expected that BB situations, once started in a route, will continue as long as the frequency remains high. Assuming the worst scenario $ET_c = f_{c,c+1}$, and assuming that there are u trips with the same frequency in that route since the beginning of bus bunching situations, there will be in the end $\frac{ET_c \times u}{2}$ extra time spent.

The cost of such situation is easy to calculate since there is an estimation of the cost per bus and per driver for each extra minute. This is of course, an upper bound for the real operational cost of bus bunching situations in terms of buses and drivers' duties. These operational costs could then be added to the benefits in terms of passengers travel time savings (based on the value-of-time estimations [73]) to assess the overall benefits from implementing a framework for preventing BB. In addition to this tangible effects, the impact of BB on service image and hence attractiveness and ridership could be assessed using satisfaction surveys in order to support the decision

making process.