Modeling dwell time in a data-rich railway environment: with operations and passenger flows data

Rémi Coulaud, Christine Keribin, Gilles Stoltz

Modeling dwell time in a data-rich railway environment: with operations and passenger flows data

Rémi Coulaud[1,2] – Christine Keribin[1] – Gilles Stoltz[1]

[1] Université Paris-Saclay, CNRS, Laboratoire de mathématiques d'Orsay, 91405, Orsay, France
{christine.keribin, gilles.stoltz}@universite-paris-saclay.fr

[2] Transilien, SNCF Voyageurs, 12 rue Jean Philippe Rameau, 93220, Saint-Denis, France
remi.coulaud@sncf.fr

August 25, 2022

Abstract

We model dwell times for trains subject to a possibly dense timetable based on a rich data set containing both railway operations variables and passenger flows variables, which is rare in the literature. Another distinguishing feature of our modeling consists of building a single statistical model for actual dwell times at all stations and in all contexts, not just in constrained situations like late arrivals or not just for some minimum dwell time. These models are fully data-driven and stem from either linear regressions with multiplicative effects or machine-learning methods like random forests, both carefully tuned on training data sets. While railway operations variables remain key for the modeling of dwell time, we are able to characterize the added value of passenger flows variables. Overall, they lead to an average reduction of the global modeling error by about 0.5 s, with up to 5 s – 10 s average improvements in challenging situations consisting, e.g., of late arrivals or associated with high passenger affluence. We also study which are the most influential variables among the available operations and passenger flows variables, and we do so globally and by regime of punctuality: for instance, passenger flows variables, and in particular, the passenger affluence at the critical door, are the most influential variables for trains suffering a late arrival, while the scheduled dwell time and the deviation to the scheduled arrival time are the most important variables for early trains.

Keywords: dwell time; timetables; modeling; passenger flows; machine-learning methods (linear regression, random forests, gradient boosting with trees)

1. Introduction and literature review

We model dwell times for trains subject to a possibly dense timetable (up to 24 trains per hour during peak hours) in the greater Paris area (SNCF operator). We do so based on two sets of variables: railway operations and timetable (scheduled dwell time, deviation to scheduled arrival time, train length, etc.), on the one hand; passenger flows (numbers of alighting and boarding passengers, occupancy rate), on the other hand. We consider two railway lines, one significantly more dense than the other, but with a common point: the vast majority of their trains is equipped with automatic passenger counting (APC) device at each door. We may therefore use the breakdown of alighting and boarding numbers by door, with a particular interest on the critical door.

Only few earlier references could use such a combination of variables based on railway operations and on passenger flows. Among these, Cornet et al. [2019] rely on similar data (same greater Paris area, same SNCF operator) and model some excess dwell time with respect to some minimum dwell time (see below), solely based on passenger flows and not using the available railway operations variables.

Also, Palmqvist et al. [2020] model dwell times in a setting with a more flexible and less precise timetable (main line trains in Sweden), relying on passenger flows (alighting, boarding, and crowding factor) and on the deviation to scheduled arrival time; their contribution, however, cannot leverage a variable like the scheduled dwell time y^{theo}, as the latter equals a single value of 42 s for all stations and trains. Thus, the quality of railway operations data prevents a direct comparison of their results to ours; in particular, they had not exhibited any effect of the deviation to scheduled arrival time on dwell time, which is, on the contrary, one of the main determinants of our models for dwell time.

Our approach is to build a single statistical model for dwell times at all stations and in all contexts, which the literature does not often offer. A key variable to be considered to that end is the regime of punctuality of a train at a given station (early arrival, i.e., arrival before the theoretical arrival time; late arrival, i.e., arrival after the theoretical departure time; arrival on time, i.e., arrival between the theoretical arrival and departure times). An important note at this stage is that we directly tackle the dwell time (the difference between the departure and the arrival times), and not some notion of "minimum" dwell time (given, e.g., by the alighting and boarding time, or by restricting the attention to the dwell time in constrained situations like late arrival).

We process the variables described above using linear regressions with or without interactions, as well as standard machine-learning methods (random forests, gradient boosting with trees, neural networks), as in Kecman and Goverde [2015]. Models inspired by the latter will form our benchmark, as they tackle dwell times in all situations, including early arrivals—which most references do not offer. As in Kecman and Goverde [2015], these benchmark models will be built solely on operations variables: with our data set we will then be able to characterize the added value of passenger flows variables to model dwell time in a railway context. In particular, we want to determine when and how much passenger flows impact railway operations.

We now detail several streams of the literature alluded at in the overall view provided above.

Dwell time modeling solely based on railway operations and timetable constraints. On a data set of railway circulation between the Hague and Rotterdam with scheduled stops, Hansen et al. [2010] exhibited a piece-wise linear relationship between dwell time and arrival delay: trains that are early or on time experience average dwell times that decrease with the earliness factor Δa, while the average dwell times of trains out of schedule are independent of how late these trains are. Of course, an explanation is that train drivers must wait the theoretical departure time even if the alighting and boarding process is over. We obtain a similar relationship on our data set, see Figure 3. Kecman and Goverde [2015] also consider data collected on trains circulating between the Hague and Rotterdam: their scheduled dwell times y^{theo}, deviations Δa to scheduled arrival time, and train types. These variables are also available on our data set and we formally define them in Section 2. They process these variables using linear regression-type methods or random forests and note that the thus constructed models for dwell times should still be improved.

Of interest is also the literature that rather aims to forecast dwell times, e.g., in some auto-regressive manner by considering past dwell times as features (at the same station for earlier trains or at earlier stations for the same train). We may cite Pritchard et al. [2021] for a UK railway network, though they only discuss delayed trains. (See also Li et al. [2016], for a Dutch railway network without a strict theoretical departure time at all stops.)

Modeling of (lower bounds on) dwell time based on passenger flows. The impact of passenger flows on dwell time was first and mostly studied for transportation means without a strict theoretical departure time, like bus, metro or light railway (as these passengers flows are then the only source of information to model dwell time). The seminal work of Levinson [1983] for buses exhibited an affine relationship between dwell time at the bus scale and passenger affluence, i.e., the sum $A + B$ of the numbers A of alighting and B of boarding passengers. Lin and Wilson [1992] for light railway in Boston and Puong [2000] for metro also in Boston (MBTA Red line) studied a multiple linear re-

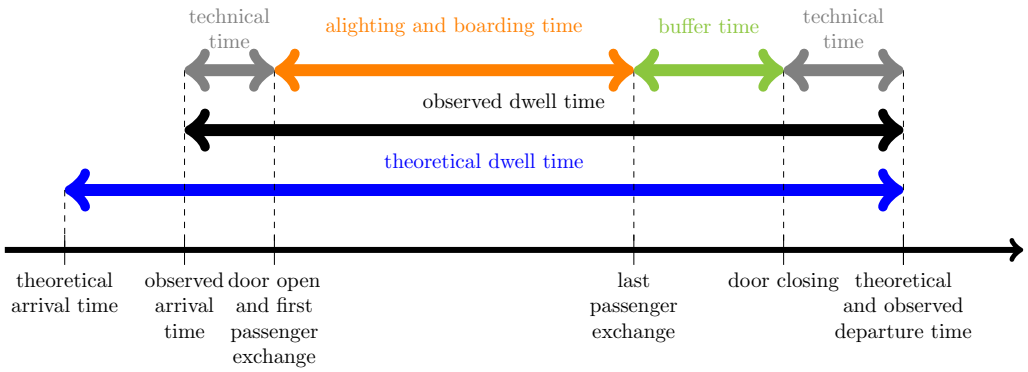


Figure 1: Decomposition of the total dwell time in railway context.

gression modeling with variables A and B considered separately and together with a crowding factor C. All these references were based on small-scale data obtained by human observations.

For the mass transit modes described in the previous paragraph, the dwell time equals the time for alighting and boarding (also known as the passenger exchange time—depicted in orange in Figure 1), i.e., the time between the first passenger exchange and the last passenger exchange, plus a technical time around the latter (e.g., to open and close the doors and to arrive or leave the stop—in grey in Figure 1). In a railway context with strict theoretical departure time, the dwell time contains a third component: a buffer time (in green in Figure 1), corresponding to some additional waiting time (till the strict theoretical departure time) when the train is early. This buffer exists by design, as some operational margin is usually added when timetables are conceived, for the sake of robustness. The literature thus rather focused on some lower bounds on the dwell time, or on the dwell time in constrained situations like late arrivals when there is no buffer time.

Among them, Buchmüller et al. [2008] studied the alighting/boarding time only for train stops without theoretical departure time constraints. Pedersen et al. [2018] and Medeossi and Nash [2020] reduced their attention to delayed trains (for which it is essentially assumed that their dwell times equal the alighting/boarding time plus a technical time, as in the case of buses and metros). The intuition behind the descriptive study by Pedersen et al. [2018] was indeed that passenger flows variables should be useful in these situations. Finally, Cornet et al. [2019] introduce some concept of empirically minimal dwell time, which they then model (essentially in some affine way). Their concept stems from running a PCA on the dwell time based on the numbers A and B of alighting and boarding passengers and the load L of the train; it turns out that the scatterplot of dwell time on the first principal component of this PCA reveals an affine lower bound.

We will propose a complete modeling of dwell time (i.e., for all stations and all trains) using passenger flows variables on top of railway operations variables. In the presence of a strict theoretical departure time, passenger flows variables provide useful additional information for dwell time modeling on top of railway operations variables (which remain the most critical variables to be used).

A specific discussion of passenger flows by door. It is intuitively clear (and was later demonstrated) that the alighting/boarding time discussed above depends on the passenger affluences $A^i + B^i$ by door i and not only on the total passenger affluence $A + B$. However, these passenger affluences $A^i + B^i$ are not uniform at all and strongly depend, for each station, on the closeness to the entry/exit of the platform (see the studies by Wirasinghe and Szplett [1984] and Wiggenraad [2001]). Yet, most data sets with passenger flows measure them only at the train scale and not at the door scale; their treatment then has to rely on an unrealistic assumption of uniform distribution of passenger affluence by door, i.e., $A^i + B^i = (A+B)/I$, where I is the number of doors. For recent examples, see Palmqvist et al. [2020] and Medeossi and Nash [2020]. (We note that Wirasinghe and Szplett [1984] proposed a theoretical model based on Gumbel's distribution for boarding numbers by door based on the location

of exit/entry platforms and the number of doors.)

On the contrary, our data set is richer than the one of Cornet et al. [2019] as it also contains door-by-door measures A^i and B^i of alighting and boarding numbers. We may then define the critical passenger affluence M, which is the maximum of the $A^i + B^i$ over the doors i, and see its added value on the modeling. If there is some (which is what we will show), then, somehow, it is proven that the assumption of uniform distribution of passenger affluence by door is unrealistic. However, as discussed in Section 5 based on the survey by Kuipers et al. [2021], there is room for further exploration of ways for defining critical passenger flows.

We cannot define a meaningful notion of crowding factor at door scale as the trains considered have corridor connections between coaches. We note that in a different context with no timetable (Beijing subway Line 13) and thus for a modeling solely based on passenger flows, Chu et al. [2015] already modeled the dwell time (equal to alighting/boarding time plus a fixed technical time in this context) based on boarding numbers per door B^i, together with global alighting numbers A and crowding factor C (which they turn into per-door quantities by diving by the number I of doors, i.e., using the unrealistic assumption of uniform distribution). Their data set was however of small scale (it was obtained by human observations).

Outline of the article

We describe the available data set and the railway context in Section 2: as discussed above, unlike most previous studies in the literature, it offers both railway operations variables and passenger flows variables. We then explain in Section 3 which machine-learning methods we consider to build, in a data-driven way, models for dwell time that can be used for all stations, all working days, all hours, and all trains. The modeling performance obtained by these models is discussed in Section 4, both at a global and at a "local" level. We summarize our conclusions in Section 5.

2. Methodology: description of the data set

We consider a suburban railway network located in the Greater Paris area, and operated by Transilien SNCF. More precisely, we are interested in two different branches of lines H and L, featuring respectively 13 stations (11 without origin/terminus) and 11 stations (9 without origin/terminus); see Figure 2. We picked them because they are completely or almost completely run with Z50000-type rolling stocks equipped both with on-train monitoring recorder (OTMR) systems, which measure speed, arrival and departure times more precisely than track circuits, and with an automatic passenger counting (APC) system, which measures, for each door of the rolling stocks, the numbers of passengers boarding and alighting at each stop. Z50000-type rolling stocks on lines H and L are composed of 8 and 7 communicating coaches, respectively. For both lines, the mean seating and total capacities by coach equal respectively 59 seats and 119 passengers. The doors width is 1.96 m. We are primarily interested in line L and provide a study of line H in Appendix C.2; we explain in depth at the end of this section why we do so.

The data set spans 18 months, from March 15, 2018 to September 16, 2019. Each daily train ride comes with a unique ID, which we will refer to as the train ID. Three primary keys will therefore be used to refer to individual data points: the train number k, station s and day d; see Table 1. We merge the two data sources (OTMR data and APC data) by matching the triplets (k, s, d). We keep all triplets present in both data sources and delete the other ones. We do not impose further restrictions, like the availability of all triplets (k, s', d) for a given day d and a given train ride k when s' spans the set of stations. The further pre-processing steps carried out are described below.

Description of the variables. The variables initially available for each triplet (k, s, d) are summarized in Table 2. Table 3 lists the variables created based on the ones of Table 2.

Table 1: Primary-key variables.

Variable	Notation
Train number	k
Station	s
Day	d

Table 2: Railway operations variables (*top and middle parts of the table*, lower case) and passenger flow variables (*bottom part of the table*, upper case).

Variable	Domain and units	Notation
Variable of interest		
– Observed dwell time	$\{0\text{ s}, 2\text{ s}, \dots, 180\text{ s}\}$	$y^{\text{obs}}_{k,s,d} = d^{\text{obs}}_{k,s,d} - a^{\text{obs}}_{k,s,d}$
Railway operations [Timetable data]		
– Theoretical (scheduled) arrival time	10 s steps	$a^{\text{theo}}_{k,s,d}$
– Theoretical (scheduled) departure time	10 s steps	$d^{\text{theo}}_{k,s,d}$
– Theoretical (scheduled) dwell time	$\{0\text{ s}, 10\text{ s}, \dots, 180\text{ s}\}$	$y^{\text{theo}}_{k,s,d} = d^{\text{theo}}_{k,s,d} - a^{\text{theo}}_{k,s,d}$
Railway operations [OTMR data]		
– Observed arrival time	2 s steps	$a^{\text{obs}}_{k,s,d}$
– Observed departure time	2 s steps	$d^{\text{obs}}_{k,s,d}$
– Capacity (maximal passenger load)	{720; 922; 1,520; 1,844}	$c_{k,d}$
– Type	{single, double}	$t_{k,d}$
Passenger flows [APC data]		
– Alighting (number of passengers alighting)	$\{0, 1, 2, 3, 4, \dots\}$	$A_{k,s,d}$
– Boarding (number of passengers boarding)	$\{0, 1, 2, 3, 4, \dots\}$	$B_{k,s,d}$
– Load of the train after departure	$\{0, 1, 2, 3, 4, \dots\}$	$L_{k,s,d}$

Table 3: Processed variables (with the same breakdown as in Table 2).

Variable	Domain and units	Notation
Railway operations		
– Way	{0,1}	w_k
		$w_k = 1$ *if train* k *goes from Paris to suburbs,* $= 0$ *from suburbs to Paris*
– Deviation to scheduled arrival time	$[-600\text{ s}, 600\text{ s}]$	$\Delta a_{k,s,d} = a^{\text{obs}}_{k,s,d} - a^{\text{theo}}_{k,s,d}$
– Regime of punctuality	{1, 2, 3}	$z_{k,s,d}$
		$= 1$ *if train is early,* $a^{\text{obs}}_{k,s,d} < a^{\text{theo}}_{k,s,d}$;
		$= 2$ *if on time,* $a^{\text{theo}}_{k,s,d} \leqslant a^{\text{obs}}_{k,s,d} \leqslant d^{\text{theo}}_{k,s,d}$;
		$= 3$ *if late,* $d^{\text{theo}}_{k,s,d} < a^{\text{obs}}_{k,s,d}$
Passenger flows		
– Crowding factor	[0, 2]	$C_{k,s,d} = L_{k,s,d}/c_{k,s,d}$
– Passenger affluence at the critical door	$\{0, 1, 2, 3, 4, \dots\}$	$M_{k,s,d}$

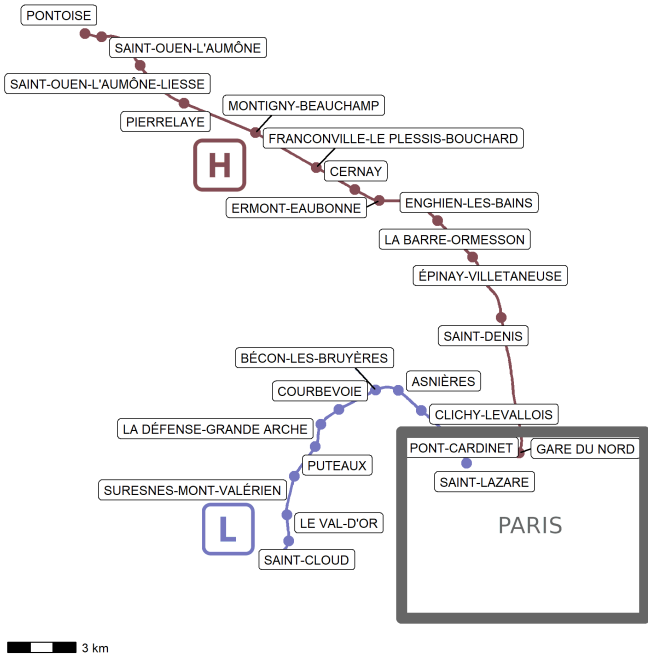


Figure 2: Branches of interest of lines H and L of the suburban railway network of the Greater Paris.

The railway operations variables of Table 2 consist first of actual (observed) and scheduled (theoretical) arrival times a^{obs} and a^{theo}, and actual (observed) and scheduled (theoretical) departure times d^{obs} and d^{theo}. Dwell times (observed and theoretical ones) are defined as the differences $y^{\text{obs}} = d^{\text{obs}} - a^{\text{obs}}$ and $y^{\text{theo}} = d^{\text{theo}} - a^{\text{theo}}$. The variable of interest is the observed dwell time y^{obs}. All these variables are indexed by triplets (k, s, d). Two final variables are only indexed by (k, d) as they only depend on the train rides, not on the specific stations: the capacity c of the rolling stocks (the maximal passenger load allowed) and their types t. The type is "single" for single-unit trains and "double" for double-unit trains. The latter are mostly used during rush hours to increase capacity. Scheduled times were obtained from the timetables while other railway operations variables were picked in the OTMR data set.

Based on the variables just described, we may compute three other railway operations variables described in Table 3. The way w (that only depends on the train number k, i.e., on the ride) indicates whether the train goes from Paris to its suburbs, or from a suburban area to Paris[1]. The deviation to the scheduled arrival time Δa is the difference $a^{\text{obs}} - a^{\text{theo}}$ between the actual arrival time a^{obs} and the scheduled one a^{theo}. Three situations may actually arise in terms of punctuality, and this leads to a final, categorical, variable called "Regime of punctuality" and denoted by z. Early trains, i.e., trains for which $a^{\text{obs}} < a^{\text{theo}}$, will be tagged with $z = 1$. Late arrivals are tagged with $z = 3$ and will refer to trains arriving after the scheduled departure time, i.e., for which $a^{\text{obs}} > d^{\text{theo}}$. (By definition, trains with a late arrival are not tied anymore by the constraint of not leaving before the scheduled departure time.) The third category $z = 2$ corresponds to trains on time, for which $a^{\text{theo}} \leqslant a^{\text{obs}} \leqslant d^{\text{theo}}$.

We only use some of the passenger flows variables available. Indeed, the APC data set reports the numbers of passengers alighting and boarding for each train at each station, globally (variables A and B) and for each door i (variables A^i and B^i). All these variables are indexed by triplets (k, s, d). The values available in the APC data set are not raw data but were obtained after some pre-processing

[1]This is an important variable in the Greater Paris area: at morning peak hours, trains from the suburbs to Paris are crowded and suffer more frequently from delays, while trains from Paris to the suburbs circulate in a smoother fashion. In the afternoon peak hours, the situation is the opposite one.

ensuring consistency (e.g., total numbers A and B are the sums of the by-door quantities A^i and B^i; the sums of the boarding numbers along the ride equal the sums of the alighting numbers). Such a pre-processing is required because of the measurement noise due to the infra-red sensor.

To avoid considering too many variables, we only use the total numbers A and B of alighting and boarding passengers (Table 2), as well as the passenger affluence at the critical door, defined as the maximal number, over the I doors, of alighting and boarding passengers at a given door i:

$$M = \max\left\{A^i + B^i : i = 1, \dots, I\right\}. \tag{1}$$

The passenger affluence at the critical door M is thus a processed variable (Table 3). A second processed variable is the crowding factor $C = L/c$, defined as the ratio between the load L and the maximal capacity c. We observe some values of C larger than 1 in the data set.

All in all, our data set is a unique combination of railway operations variables (typically accessible) with rich passenger flows variables (seldom available). The closest data set in the literature is the one of Cornet et al. [2019], which however does not contain by-door measures of passenger affluence.

Modeling vs. prediction. The focus of the present article is only on modeling dwell time based on the explanatory variables described above. The passenger flows variables are available in real time (i.e., right after the train leaves a station) while the railway operations variables are only known with some delay (they are not transmitted in real time). To move from modeling to prediction we would need to predict passenger flows variables for the next station and know the railway operations variables in real time (e.g., know the deviation Δa to scheduled arrival when the train stops and the passenger exchange starts taking place). It turns out that the APC data set actually contains some railway operations variables, measured in real time, but they are less reliable than the OTMR measurements. In any case, we would need predictions for passengers flows. This is why the focus of the present article is only on modeling (i.e., explaining the determinants of dwell time) and not on forecasting.

Further pre-processing of the data / Data volume. On top of the pre-processing described above, which consisted of keeping only observations and variables relative to triplets (k, s, d) present in both data sources (OTMR and APC), we performed some data cleaning. First, we deleted triplets (k, s, d) corresponding to anomalous situations: when the observed dwell time $y^{\text{obs}}_{k,s,d}$ is longer than 180 seconds (as Cornet et al. [2019], Dueker et al. [2004] did) or when current cumulative delays on the ride are larger than 10 minutes.

Doing so, we get more than 350,000 observations for line L and 416,000 for line H.

Railway contexts: line L is more important than line H. In this article we only report results for line L but discuss line H in Appendix C.2. We explain here the several reasons why we favor line L over line H in our study.

First, the passenger flows on line L are more varied than on line H. On line H, most of the passenger flows take place at terminus, while on line L, there exist major intermediate stations (like La Défense Grande Arche) also generating major passenger flows. The average passenger volumes vary from 1,300 to 37,000 passengers per day on the considered branch of line L.

Second, the railway operations are more challenging on line L than on line H. On the one hand, the traffic on line L is more dense than the one of line H for stations distant from Paris, on peak hours: typically, a train every 5 minutes on line L versus every 15 minutes on line H. (For stations close to Paris, the density is similar, with about 22 to 24 trains per hour, that is, a train every 2 to 3 minutes.) On the other hand, lines L and H also differ in terms of punctuality, with a greater variety of situations for line L: most of the train rides on line H end up being on time or late, while there is also a significant fraction of early train rides on line L on top of on-time and late train rides.

Finally, boarding volumes on line L are sufficiently higher than on line H to result in larger crowding factors, despite the higher density of trains.

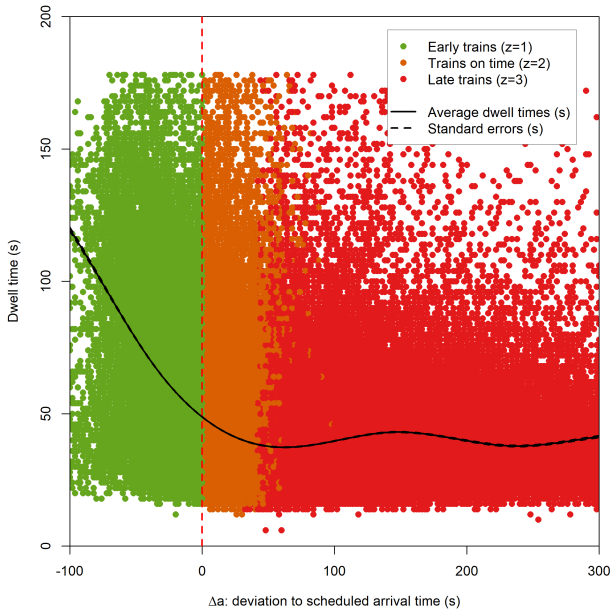


Figure 3: Observed dwell times (y–axis, seconds) by deviations Δa to scheduled arrival times (x–axis, seconds); we also report average dwell times and standard errors thereof, based on some generalized additive modeling. (The corresponding lines are extremely close to each other.) Three regimes are considered: early trains, trains on time, late trains.

Figure 3 depicts the observed dwell times, as well as the averages thereof, on the considered branch of line L, by deviations Δa to scheduled arrival times. The averages and standard errors were obtained by a generalized additive method modeling based on 10 cubic splines, see Wood [2006]. We build confidence intervals around the averages of half-widths ± 2 standard errors.

3. Methodology: regression models and machine-learning methods

We model the observed dwell times $y^{\text{obs}}_{k,s,d}$ as a stochastic function of *some* of the variables described in Tables 2 and 3, namely,

$$y^{\text{obs}}_{k,s,d} = f(s,\, w_k,\, t_{k,d},\, y^{\text{theo}}_{k,s,d},\, \Delta a_{k,s,d},\, z_{k,s,d},\, A_{k,s,d},\, B_{k,s,d},\, C_{k,s,d},\, M_{k,s,d}) + \varepsilon_{k,s,d}\,, \tag{2}$$

where f is some deterministic function and the additive residual terms $\varepsilon_{k,s,d}$ are random variables (assumptions thereon will depend on each method used, see below). We justify in Section 3.1 below the choice of the variables used in Equation (2).

We are interested in some statistical modeling and do not propose simulation or probabilistic models (as did D'Acierno et al. [2017] or Cornet et al. [2019]). Also, we model directly $y^{\text{obs}}_{k,s,d}$, and not $y^{\text{obs}} - y^{\text{theo}}$, as we want to separate strictly the respective information provided by passenger flows variables and railway operations variables. See Appendix C.1 for more details and a report of the performance obtained by rather modeling $y^{\text{obs}} - y^{\text{theo}}$.

3.1. Justification of the variables used

First, for each Transilien network branch, the combination of the station s and the way w_k indicates on which specific platform the train will stop. This is important in light of studies like the one by Daamen et al. [2008], who confirmed the major impact of platform design (stepping gap, height difference, etc.) on the alighting and boarding time.

Now, out of the many railway operations variables available, we only use $y^{\text{theo}}_{k,s,d}$, $\Delta a_{k,s,d}$, $z_{k,s,d}$, $t_{k,d}$. We do so because we want to build a model that can be easily grasped. First, the scheduled dwell time y^{theo} of course provides some benchmark on the expected dwell times; this piece of information is typically used in modelings, in some direct or indirect way; see, among others, Kecman and Goverde [2015] and Li et al. [2016]. We already explained that Hansen et al. [2010] showed how important the deviation to scheduled arrival time Δa is to explain the dwell times, and Figure 3 illustrated it. We build regimes of punctuality z based on Δa (see Table 3) to isolate unconstrained dwell times (for late trains, $z = 3$) from dwell times constrained by the scheduled departure times. In particular, we expect that early trains which do not face too high a passenger affluence need to wait till the scheduled departure time and hence, have an observed dwell time equal to $y^{\text{theo}} + |\Delta a|$, the scheduled dwell time plus how early they were. On the contrary, we expect that drivers of late trains will try to shorten dwell times as much as possible.

Finally, the type t of train is also important: it provides an indirect idea of the expected passenger affluence, as double-unit trains are only used when necessary; this idea is inspired from Kecman and Goverde [2015]. It may also help because double-unit trains and single-unit trains occupy different shares of the platform, and we already mentioned how important the design of the platform is. However, we chose not to consider the other railway operations variables, that should either be irrelevant (scheduled departure time, scheduled and observed arrival times should not convey any information beyond what is already contained in Δa and y^{theo}) or be future variables (the observed departure time is basically what is to be modeled).

As far as passenger flows variables are considered, we consider them all except the load L of the train, as the latter only has a meaning relative to the train capacity—hence the crowding factor C.

Remark: no auto-regressive modeling. Our aim is to model dwell times based on the current context (state of railway operations, passenger affluence, etc.) and determine which elements of this context have the most important influence on dwell time. Our aim is not to forecast dwell times. Therefore, we do not consider auto-regressive-type models, i.e., we do not include variables like $y^{\text{obs}}_{k-1,s,d}$ or $y^{\text{obs}}_{k,s-1,d}$ in the modeling of $y^{\text{obs}}_{k,s,d}$. See Li et al. [2016] and Pritchard et al. [2021] for such modelings.

Subsets of variables: RO, PF, M. As we want to determine which variables are most influential, we group them in two groups and a half. We always use s and w_k and cluster the rest of the variables into

- Railway operations variables [short-hand notation "RO"]: $y^{\text{theo}}_{k,s,d}$, $\Delta a_{k,s,d}$, $z_{k,s,d}$, $t_{k,d}$;
- Passenger flows variables [short-hand notation "PF"], not taking into account the passenger affluence at the critical door: $A_{k,s,d}$, $B_{k,s,d}$, $C_{k,s,d}$;
- Passenger affluence at the critical door [short-hand notation "M"]: $M_{k,s,d}$.

We will actually run our methods with s, w_k and either just RO variables, or just PF variables, or RO+PF variables, or RO+PF+M variables.

3.2. One model for all stations, all working days, all hours, and all trains

We restrict our attention to working days (i.e., Mondays to Fridays that are not public holidays nor belong to school holidays). We do so because we want to assess the impact of passenger flows on dwell time, and these flows are limited on non-working days. Our second aim is to build models suitable for all stations, all working days, at all hours and for all trains simultaneously, using only variables s (the station) and w_k (the way of train k) to locally adapt the model.

We however do not try to provide a general model that would work for all train networks (as in Harris and Anderson [2007] and Li et al. [2016]), but rather provide a general methodology to adjust

specific dwell-time models for each (sub)network suitably equipped in terms of monitoring devices (APC and OTMR ones).

Linear regression models (see, among others, Lam et al. [1998], Harris and Anderson [2007], Palmqvist et al. [2020]) are a popular such general methodology, that leads to easily interpretable models. Even with linear regression models, some non-linear modeling may be achieved by considering multiplicative effects, which we will do. Machine-learning models were later considered (see, among others, Kecman and Goverde [2015]) to improve the accuracy of the modeling based on linear regressions, at the cost of building black-box models which are highly non-linear per design.

We describe the linear regression models considered in Section 3.3, and then mention the machine-learning methods considered: random forests and gradient boosting in Section 3.4, and neural networks in Section 3.5. We provide concise descriptions of these machine-learning methods in Appendix A and refer interested readers to Hastie et al. [2009] for deeper expositions. Finally, we explain in Section 3.6 how to tune these methods (on a train set) and evaluate them in a fair way (on a test set).

3.3. Linear regression models (with additive or multiplicative effects)

The simplest version of linear regression models uses an affine function f in Equation (2). In f, the quantitative variables, namely, $y^{\text{theo}}_{k,s,d}$ and $\Delta a_{k,s,d}$ when RO variables are considered, $A_{k,s,d}$, $B_{k,s,d}$, $C_{k,s,d}$ when PF variables are considered, and $M_{k,s,d}$ for the M variable, are each associated with slope coefficients denoted by $\beta^{(y)}$, $\beta^{(\Delta a)}$, $\beta^{(A)}$, $\beta^{(B)}$, $\beta^{(C)}$, and $\beta^{(M)}$, respectively. As for the categorical variables, namely, s and w_k in all cases, and $z_{k,s,d}$ and $t_{k,d}$ when RO variables are considered, we include them by considering $K-1$ indicator variables, where K denotes the number of modalities taken. For instance, we denote by S the number of stations and index them by $1, \ldots, S$; we take the last station as a reference modality and the regression function f thus features $S-1$ coefficients $\beta^{\text{station}}_{s'}$, where $s' \in \{1, \ldots, S-1\}$. Similarly, for t and w, which both only take two values, we take the modalities "single" and $w = 0$ (from suburbs to Paris) as reference values, and the regression function f features the coefficient β^{type} and β^{way}. Finally, for the variable z which take three modalities, we pick $z = 2$ (trains on time) as a reference value and thus have two coefficients β^{early} and β^{late} for inclusion in f. We denote the global intercept by β^0. All in all, with the simultaneous consideration of the RO, PF, and M variables, we use in Equation (2)

$$
\begin{aligned}
&f(s, w_k, t_{k,d}, y^{\text{theo}}_{k,s,d}, \Delta a_{k,s,d}, z_{k,s,d}, A_{k,s,d}, B_{k,s,d}, C_{k,s,d}, M_{k,s,d}) \qquad (3)\\
&= \left.\begin{aligned} &\beta^0 + \beta^{\text{way}} \mathbb{1}_{[w_k=1]} + \sum_{s'=1}^{S-1} \beta^{\text{station}}_{s'} \mathbb{1}_{[s=s']} \end{aligned}\right\} \text{ in all cases}\\
&\quad \left.\begin{aligned} &+ \beta^{(\Delta a)} \Delta a_{k,s,d} + \beta^{(y)} y^{\text{theo}}_{k,s,d} + \beta^{\text{type}} \mathbb{1}_{[t_{k,d}=\text{double}]} + \beta^{\text{early}} \mathbb{1}_{[z_{k,s,d}=1]} + \beta^{\text{late}} \mathbb{1}_{[z_{k,s,d}=3]} \end{aligned}\right\} \text{ RO variables}\\
&\quad \left.\begin{aligned} &+ \beta^{(A)} A_{k,s,d} + \beta^{(B)} B_{k,s,d} + \beta^{(C)} C_{k,s,d} \qquad + \beta^{(M)} M_{k,s,d} \end{aligned}\right\} \text{ PF+M variables}
\end{aligned}
$$

If we only use some of these variables, we suppress some terms in the equation above (e.g., the second line if we only use the PF+M variables; or the term $\beta^{(M)} M_{k,s,d}$ if we use the RO+PF variables).

Linear regression with additive effects. We call the model above the linear regression with additive effects. It features $S+1$ coefficients in all cases, plus 5 coefficients when RO variables are included, 3 when PF variables are used, and 1 for the M variable, respectively. This leads to the numbers of coefficient stated in the first line of Table 4.

Multiplicative effect of Δa by z. To take into consideration the special relation between the dwell time and the deviation to scheduled arrival time (see Figure 3), we provide a different affine modeling in terms of $\Delta a_{k,s,d}$ for each value of $z_{k,s,d}$. Put differently, instead of a single slope coefficient $\beta^{(\Delta a)}$

Table 4: Numbers of coefficients of the various linear regressions considered, for line L (for which there are $S = 10$ stations).

	Variables used			
	PF	RO	RO+PF	RO+PF+M
Additive effects	14	16	19	20
Multiplicative effect of Δa by z	14	18	21	22
Multiplicative effects by (s, w, z)	$\geqslant 180$	$\geqslant 120$	$\geqslant 300$	$\geqslant 360$

in front of $\Delta a_{k,s,d}$, we provide a breakdown by punctuality $z_{k,s,d} \in \{1, 2, 3\}$ and use three different slope coefficients $\beta_1^{(\Delta a)}$, $\beta_2^{(\Delta a)}$, $\beta_3^{(\Delta a)}$. We do this on top of setting different intercept levels through the consideration of β^{early} and β^{late}. That is, with the simultaneous consideration of the RO, PF, and M variables, we use in Equation (2)

$$
\begin{aligned}
& f(s, w_k,\, t_{k,d},\, y^{\text{theo}}_{k,s,d},\, \Delta a_{k,s,d},\, z_{k,s,d},\, B_{k,s,d},\, A_{k,s,d},\, C_{k,s,d},\, M_{k,s,d}) \qquad (4)\\
&= \beta^0 + \beta^{\text{way}} \mathbb{1}_{[w_k=1]} + \sum_{s'=1}^{S-1} \beta^{\text{station}}_{s'} \mathbb{1}_{[s=s']} \quad \Big\} \text{ in all cases}\\
&\quad + \sum_{z\in\{1,2,3\}} \mathbb{1}_{[z_{k,s,d}=z]}\, \beta_z^{(\Delta a)}\, \Delta a_{k,s,d} \quad \Big\} \text{ RO variables, new part: interaction between } \Delta a \text{ and } z\\
&\quad + \beta^{(y)}\, y^{\text{theo}}_{k,s,d} + \beta^{\text{type}} \mathbb{1}_{[t_{k,d}=\text{double}]} + \beta^{\text{early}} \mathbb{1}_{[z_{k,s,d}=1]} + \beta^{\text{late}} \mathbb{1}_{[z_{k,s,d}=3]} \quad \Big\} \text{ RO variables, no change}\\
&\quad + \beta^{(A)}\, A_{k,s,d} + \beta^{(B)}\, B_{k,s,d} + \beta^{(C)}\, C_{k,s,d} \qquad + \beta^{(M)}\, M_{k,s,d} \quad \Big\} \text{ PF+M variables}
\end{aligned}
$$

We call the model above the linear regression with a multiplicative effect of Δa by z. When RO variables are considered, it contains two additional coefficients with respect to the model with additive effects and does not differ from the latter when RO variables are omitted; see the second line of Table 4.

Additional multiplicative effects. We may have the slope coefficients, as well as the intercepts, vary by pairs (s, z) or even, triplets (s, w, z) to locally tailor the model to the stations and to the regime of punctuality; i.e., with PF variables, the regression function f would, for instance, feature terms like

$$
\sum_{\substack{s'\in\{1,\dots,S\}\\ w\in\{0,1\}\\ z\in\{1,2,3\}}} \mathbb{1}_{\left[\begin{smallmatrix} s=s',\\ w_k=w,\\ z_{k,s,d}=z \end{smallmatrix}\right]} \beta^{(A)}_{s,w,z}\, A_{k,s,d} + \sum_{\substack{s'\in\{1,\dots,S\}\\ w\in\{0,1\}\\ z\in\{1,2,3\}}} \mathbb{1}_{\left[\begin{smallmatrix} s=s',\\ w_k=w,\\ z_{k,s,d}=z \end{smallmatrix}\right]} \beta^{(B)}_{s,w,z}\, B_{k,s,d} + \sum_{\substack{s'\in\{1,\dots,S\}\\ w\in\{0,1\}\\ z\in\{1,2,3\}}} \mathbb{1}_{\left[\begin{smallmatrix} s=s',\\ w_k=w,\\ z_{k,s,d}=z \end{smallmatrix}\right]} \beta^{(C)}_{s,w,z}\, C_{k,s,d} \qquad (5)
$$

instead of $\beta^{(A)}\, A_{k,s,d} + \beta^{(B)}\, B_{k,s,d} + \beta^{(C)}\, C_{k,s,d}$. For multiplicative effects by triplets, we end up with models with at least $6S$ coefficients per quantitative variable considered (the total number of coefficient depending on the specific dependencies considered for the intercepts); see the third line of Table 4. We tried many formulations and all got a similar performance.

The linear models discussed in this section are reference models and are mostly of interest for the sake of comparison with more complex, machine-learning, methods, which often exhibit a better performance, at the cost of not leading to statistical models, i.e., closed-form relationships that may be interpreted. We consider two methods based on regression trees, which we discuss now, and one on neural networks, which we discuss later.

3.4. Machine-learning methods based on regression trees: random forests and gradient boosting

Machine-learning methods based on regression trees were already considered by the literature on transportation systems: Kecman and Goverde [2015] used random forests to model dwell time for trains circulating between the Hague and Rotterdam, based on similar railway operations (RO) variables as we consider in this article; Ding et al. [2016] used gradient boosting and were interested in short-term metro ridership forecasting (next 15 minutes) on three major Beijing stations. Zhang and Haghani [2015] considered both methods to forecast car travel time on a motorway section in Maryland; in their study, boosting methods slightly outperformed random forests. All three references present in details random forests and gradient boosting (and do so by first introducing regression trees). We also provide a description of our own in Appendix A, which introduces all quantities T, m, η, $\mathcal{F}$, etc., used below in the indications of how we implemented these methods.

Random forests. We implemented random forests using the R package `ranger` (see Wright and Ziegler, 2017; it is better suited to large data sets than, e.g., the `randomForest` package). It uses two parameters, `ntree` for the number of trees T and `mtry` for the number m of variables chosen at each split. Both are tuned by cross validation, see Section 3.6. The bootstrapped data samples are of the same size as the original data set.

Gradient boosting with regression trees. The specific method at hand is XGBoost by Chen and Guestrin [2016]. We use the R package `xgboost`. The XGBoost method may be finely tuned through a few dozens of parameters, including the choice of the tree set $\mathcal{F}$; we use the default values, except for the number of iterations T and the step size η (which correspond to the parameters `nrounds` and `eta`, respectively), which we tune by cross validation, see Section 3.6. It is a common choice (both in machine-learning competitions and in the transportation literature, see Ding et al., 2016) to focus mostly on these two parameters.

3.5. Feed-forward neural networks

Artificial neural networks (see Goodfellow et al., 2016) are a popular method for designing highly non-linear predictors, in all fields of science and engineering, including transportation research. They are considered in transportation research about public transports with relatively simple architectures, typically based on at most one hidden layer. For instance, Yaghini et al. [2013] used such simple neural networks to classify train delays for Iranian railways, while Amita et al. [2015] did so to predict bus running times in Dehli based on GPS data. In traffic literature more complex architectures are often considered, as did Li et al. [2018] for the forecasting of road traffic flows on two data sets from California highways. They compare two methods, a dense feed-forward neural network with two hidden layers of 256 nodes each and a more complex diffusion convolution recurrent neural network. As we face a public-transport application, we do not consider the latter method and only proceed with feed-forward neural networks.

The mentioned references all consider different architectures for their feed-forward neural networks: to a great extent, the choice of the architecture of a neural network is subjective and relies on engineering experience. However, in this work, we consider the number of hidden layers H and the number N of nodes per layer as tuning parameters (to be chosen based on data through cross validation, see Section 3.6).

The architecture considered for our feed-forward networks is depicted in Figure 4; it is composed of an input layer, of H hidden dense layers (each with N nodes), and of an output layer. The translation of this architecture into a specific modeling $f(\boldsymbol{X}_{k,s,d})$ is provided, for the sake of completeness, in Appendix A.

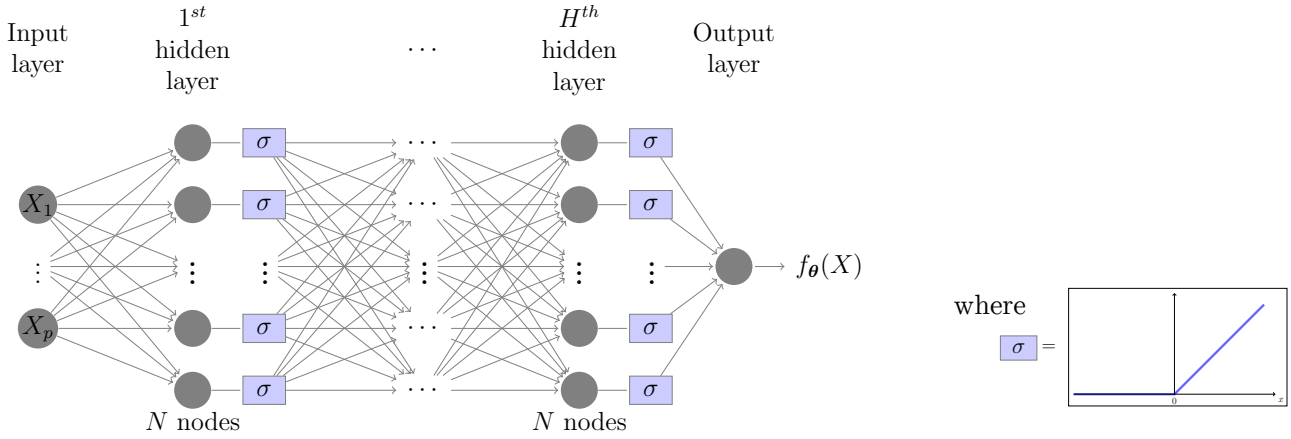


Figure 4: Architecture of the feed-forward networks considered; the σ boxes correspond to the application of the rectified linear unit (ReLU) activation function $\sigma(x) = \max\{x, 0\}$ depicted on the right.

Table 5: Grids for picking the hyperparameters (a.k.a. tuning parameters) on the train set.

Method	Hyperparameter #1	Hyperparameter #2
Random forests	$m \in \{1, 2, \ldots, 15\}$	$T \in \{1, 10, 50, 100, 500, 1000, 5000\}$
Gradient boosting with regression trees	$\eta \in \{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$	$T \in \{1, 200, 600, 1000, 2000, 6000, 8000\}$
Feed-forward neural networks	$H \in \{1, 2, 3, 4, 5, 6\}$	$N \in \{32, 64, 128, 256, 512\}$

For our implementation, we use the R package `keras` to build the architecture and the R package `tensorflow` to train the network. The model is trained with batch size 32 and mean absolute error as the loss function (see Section 3.6). We use the classical `Adam` optimizer, which is based on stochastic sampling, to compute gradients. We run 50 epochs, not more (to avoid over-fitting), not fewer (to train sufficiently the parameters).

3.6. Fair assessment of the performance: picking parameters on a train set, and evaluating performance on a test set

All methods above require some training on historical data. Fitting the coefficients of the linear regression models on such historical data is straightforward. Machine-learning techniques (random forests, gradient boosting with regression trees, and feed-forward neural networks) require a more sophisticated use of historical data: they need to pick some hyperparameters—two per method, which we recall in Table 5—and fit the model on data based on these hyperparameters.

A popular solution in statistics, already considered in transportation research by, among others, Kecman and Goverde [2015], consists in separating the data set into two subsets: a train data set and a test data set. For machine-learning methods, the train data set is used both to select hyperparameters by (5-fold) cross-validation and fit the models accordingly. To do so, our procedure consists of two passes on the train data set, a first to select the hyperparameters, based on cross-validation, and a second to fit the corresponding model. A more detailed statement of this procedure may be found in Appendix A.3.1. For linear regression models, we directly fit coefficients on the train data set. The test data set is used to evaluate the performance of the thus constructed and fitted methods. Doing so, we avoid favorable biases that would consist, for instance, of constructing and evaluating the methods

Table 6: Tuning parameters selected based on the considered sets of variables.

Methods	Pairs of parameters	Variables sets			
		PF	RO	RO PF	RO PF+M
Random forests	$(T, m) =$	$(10, 15)$	$(100, 10)$	$(500, 5)$	$(5000, 7)$
Gradient boosting with regression trees	$(T, \eta) =$	$(6000, 0.0005)$	$(8000, 0.0005)$	$(6000, 0.005)$	$(6000, 0.005)$
Feed-forward neural networks	$(N, H) =$	$(256, 5)$	$(32, 4)$	$(32, 4)$	$(64, 2)$

on the same data subset; in that case, we would be providing some in-sample error rather than an out-of-sample error.

Breakdown used. We consider a fixed 60%–40% breakdown of the data set into a train data set (data points from March 15, 2018 to March 15, 2019) and a test data set (data points from March 16, 2019 to September 15, 2019). We set the 60%–40% proportions in some arbitrary way.

A note on the hyperparameters considered. In the cross-validation procedure alluded at above, hyperparameters are selected based on grids of possible values, provided in Table 5. These grids had been determined ex ante and were constructed based on previous choices of these hyperparameters in the literature. For random forests, we built the grids around the default parameters of the `ranger` package (see Wright and Ziegler, 2017), which equal $m = \lfloor \sqrt{p} \rfloor$ for `mtry` (where p is the number of input variables considered) and $T = 500$ for `ntree`; given the values of p (see Table 4), this leads to $m = 4$ or $m = 5$. For gradient boosting with regression trees, we take the exact same grids as considered by Zhang and Haghani [2015, Section 3.2]. For the feed-forward neural networks, we built a reasonable grid based on the default values $N = 256$ units and $H = 2$ hidden layers chosen by Li et al. [2018, Annex E].

The hyperparameters selected by the (5-fold) cross-validation procedure are reported in Table 6. These are the hyperparameters we use in the rest of the article.

It turns out that on the data set considered, the performance of the machine learning methods is not too sensitive to the pairs of hyperparameters considered. More details are to be found in Appendix A.3.2, where the performance of the methods are tabulated on the grids of Table 5 and where we observe that whenever these hyperparameters are large enough, a close-to-optimal performance is reached.

4. Main results

The previous section described machine-learning methods to build data-driven models for dwell time valid for all stations, all working days, all hours, and all trains. In this section, we quantify their modeling performance, i.e., report the modeling errors described in Section 4.1, namely, the mean absolute modeling errors and the root mean squared modeling errors. We do so both at a global level (Section 4.2) and at a local level (Section 4.3), possibly also by considering an addition breakdown of the modeling performance by regimes of punctuality or passenger affluence (Section 4.4). By "global" results, we mean errors obtained by global averages over all stations, all working days, all hours, and all trains. By "local" results, we mean conditional averages of the form "average error suffered when some explanatory variable equals a given value". We of course define first more formally the concept

of "local" performance Section 4.3. We conclude by a ranking of the explanatory variables depending on their modeling influence (Section 4.5).

Additional results about local performance by the observed dwell times may be found in Appendix B.

4.1. Metrics for the assessment of performance

With the notation of Section 3, models $\widehat{f}$ are built on the train set and are evaluated on the test set $\mathcal{T}\text{est}$, whose cardinality is denoted by $N_{\mathcal{T}\text{est}}$. The mean absolute error (MAE) and root mean squared error (RMSE) of such a model $\widehat{f}$ are respectively defined by

$$\text{MAE}\big(\widehat{f}\big) = \frac{1}{N_{\mathcal{T}\text{est}}} \sum_{(k,s,d)\in\mathcal{T}\text{est}} \Big| y^{\text{obs}}_{k,s,d} - \widehat{f}(\boldsymbol{X}_{k,s,d}) \Big| \tag{6}$$

$$\text{and} \qquad \text{RMSE}\big(\widehat{f}\big) = \sqrt{\frac{1}{N_{\mathcal{T}\text{est}}} \sum_{(k,s,d)\in\mathcal{T}\text{est}} \Big(y^{\text{obs}}_{k,s,d} - \widehat{f}(\boldsymbol{X}_{k,s,d}) \Big)^2}. \tag{7}$$

No metric seems preferred in the transportation literature, and each has its own advantages: MAE summarizes best the global performance while RMSE is sensitive to large errors.

4.2. Main table: global performance

Table 7 reports the global performance for the modeling of dwell time, i.e., the MAE and the RMSE achieved on the entire test data set, of the six methods presented in Sections 3.3–3.5 run on the four possible subsets of variables described in Section 3.1.

We first comment how the modeling performance depends on the subsets of variables. Using passenger flows [PF] variables only is suboptimal, and railway operations [RO] variables seem key to achieve the best performance. We also observe that overall, using RO variables only is not as good as using RO and PF variables simultaneously, which is itself slightly outperformed by using RO and PF variables together with the M variable consisting of the passenger affluence at the critical door. The observations made above are consistent with previous observations in the literature, which deemed RO variables more important than PF variables for commuter trains (Hansen et al., 2010, Kecman and Goverde, 2015). We detail in subsequent subsections how PF variables, including the M variable, are valuable to consider on top of RO variables. This will, in particular, show the genuine interest of the PF and M variables, which, for now, seems modest on Table 7—while one could have expected a more dramatic effect based on the study by Wirasinghe and Szplett [1984].

We now comment the influence of the method. We first observe that the more complex the linear regression models, the better the performance. But linear regression models, which provide explainable relationships, exhibit suboptimal performance compared to the machine-learning methods (random forests, gradient boosting with regression trees, feed-forward neural networks), which do not offer explicit relationships and only provide black-box (highly non-linear) modelings. Among these machine-learning methods, gradient boosting with regression trees performs slightly better than random forests and feed-forward neural networks. All in all, the linear regression with a multiplicative effect of Δa by z probably offers the best trade-off, among all six methods considered, between simplicity, explainability and performance.

4.3. "Local" performance, depending on the level of explanatory variables

We now provide a more "local" study of performance: instead of reporting global measures of performance, we rather explain how performance varies as a given explanatory variable (passenger affluence, deviation to scheduled arrival time, etc.) varies. For the sake of concision, we will only consider one machine-learning method; to allow comparison to earlier results, we select random forests: Kecman

Table 7: Modeling performance for each method and each set of variables, in MAE (*left part of the table*) and RMSE (*right part of the table*). Columns indicate which variables are used (see Section 3.1): only passenger flows [PF] variables, only railway operations [RO] variables, both RO and PF variables, and all variables (RO, PF, and M, the passenger affluence at the critical door). Each line corresponds to a method to process data: linear regressions (Section 3.3), random forests and gradient boosting (Section 3.4), feed-forward neural networks (Section 3.5). Standard errors are smaller than 0.03 seconds.

Methods	MAE				RMSE			
	PF	RO	RO PF	RO PF+M	PF	RO	RO PF	RO PF+M
1. Linear regression with additive effects	13.7	10.5	10.2	10.1	18.4	14.8	14.5	14.3
2. Linear regression with a multiplicative effect of Δa by z	13.7	9.1	8.9	8.8	18.4	13.6	13.2	13.1
3. Linear regression with multiplicative effects by triplets (s, w, z)	13.3	8.8	8.3	8.3	18.0	13.2	12.6	12.5
4. Random forests	13.7	8.4	8.1	8.0	18.8	12.9	12.5	12.3
5. Gradient boosting with regression trees	12.9	8.5	8.0	7.9	17.9	13.0	12.4	12.2
6. Feed-forward neural networks	12.7	8.4	8.0	8.0	17.4	13.0	12.4	12.2

and Goverde [2015] ran random forests on RO variables, and we will be running them also on all variables (RO, PF, and M). We will refer to both instances of random forests by the short-hand notation RF–RO and RF–All.

Our main aim in this section is to highlight the added value of considering PF and M variables on top of RO variables: while the fourth line of Table 7 shows an extremely similar global performance of RF–RO and RF–All, we will demonstrate improvements in the "local" performance thereof. We first explain how we define and compute the latter.

Concept of local performance. We merely describe here how Figures 5–8 were obtained and how they measure local performance. We comment below on the gain in efficiency brought by RF–All with respect to RF–RO, in a dedicated series of paragraph.

Figures 5–8 aim to illustrate the impact of passenger affluence $A + B$ (the sum of the numbers of passengers alighting plus the ones boarding), which is to be found in x–axis, on performance for the modeling of dwell time, which is to be found in y–axis. This performance may be measured in an absolute manner (for RF–RO or for RF–All, as in the left graph of Figure 6) or in a relative manner (improvement of RF–All over RF–RO, as in the right graph of Figure 6).

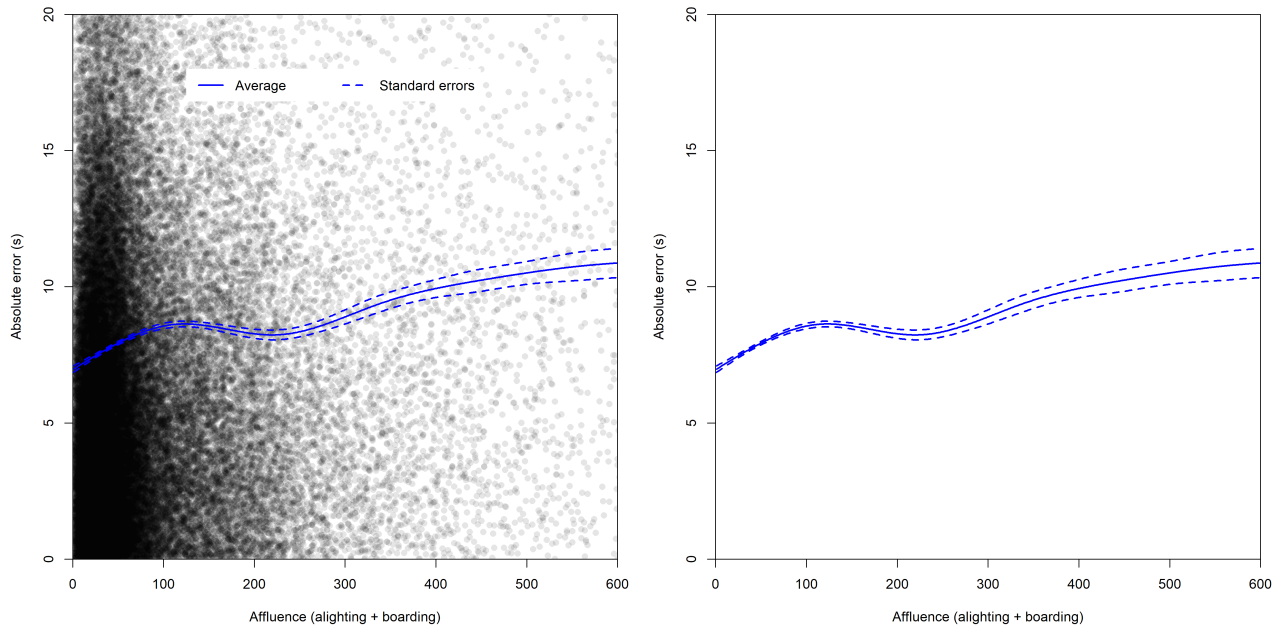


Figure 5: Left graph: scatterplot of the absolute errors on the test set for dwell time modeling by RF–All plotted against passenger affluence, together with an estimation of the associated average absolute errors (solid line), and standard errors thereof (dotted lines). Right graph: left graph without the underlying scatterplot.

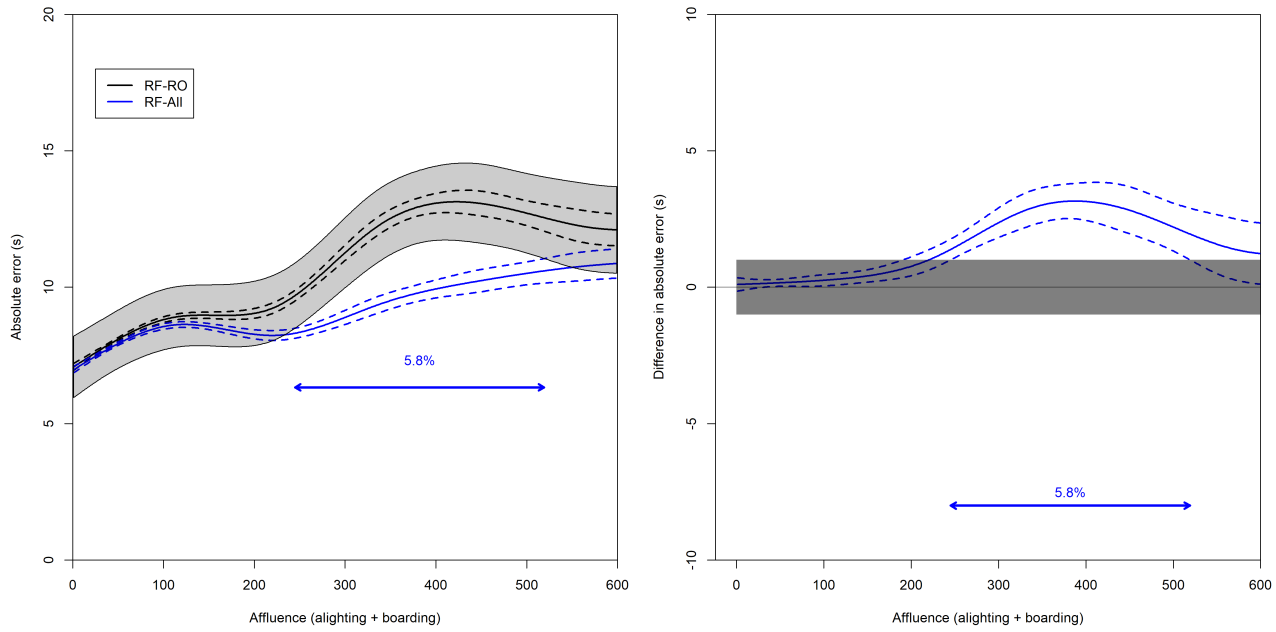


Figure 6: Left graph: average absolute error for the modeling of dwell time (y–axis) by passenger affluence (x–axis) for RF–RO (black) and RF–All (blue). Right graph: difference of these average absolute errors, between RF–RO and RF–All. The horizontal arrows indicate the data ranges where significant improvements are achieved on average by RF–All over RF–RO; the percentages below the arrows are the corresponding data shares.

We explain first how local performance is measured in an absolute manner. We fix a given method, say, RF–All. The scatterplot underlying the left graph of Figure 5 consists of the pairs

$$\left(A_{k,s,d} + B_{k,s,d},\ \left| y^{\text{obs}}_{k,s,d} - \widehat{f}(\boldsymbol{X}_{k,s,d}) \right| \right) \tag{8}$$

as (k, s, d) varies in the test set $\mathcal{T}$est. We apply the same smoothing as in Figure 3. Doing so, we obtain a curve representing the average absolute error in the modeling of dwell time by the level of passenger affluence (solid line); this average is associated with a ± 2 times standard deviation (dotted lines). The right graph of Figure 5 is simply a cleaned version of the left one, where we erased the underlying scatterplot.

Now, the representation just described may be performed for RF–RO and for RF–All: see the left graph of Figure 6, where we also added a ± 1 s tube starting from the dotted lines. This tube measures the significant improvements: as dwell time is measured with 2 s steps (see Table 2), we are only interested in average improvements larger than 1 s. We may read on the left graph the range where RF–All improves significantly over RF–RO: the range where the lower part of the tube around RF–RO is higher than the upper dotted line for RF–All. This range accounts for 5.8% of the observations, as we write on the blue arrow under the curves.

We represent this comparison in an equivalent manner on the right graph of Figure 6: the average difference by passenger affluence is the difference of the average absolute errors by passenger affluence between RF–RO and RF–All, and the associated standard deviations are the sums of the standard deviations associated with the average errors of RF–RO and RF–All. The same ± 1 s tube is depicted, around the value 0.

We may proceed similarly with square errors for the modeling of dwell time. The left graph of Figure 7 depicts the scatterplot of

$$\left(A_{k,s,d} + B_{k,s,d},\ \left(y^{\text{obs}}_{k,s,d} - \widehat{f}(\boldsymbol{X}_{k,s,d}) \right)^2 \right) \tag{9}$$

as (k, s, d) varies in the test set $\mathcal{T}$est. Average squared errors by passenger affluence and their associated ± 2 standard deviations may then be computed, exactly as in the left graph of Figure 5. The right graph of Figure 7 depicts the roots of the curves computed in the left graph of Figure 7; these root curves depict root mean square errors by the passenger affluence, associated with measures of deviations. The left graph of Figure 8 provides such root curves for RF–RO (together with a ± 1 s tube) and RF–All, while the right graph of Figure 8 is the difference between these curves, in the RF–RO minus RF–All direction.

Comments on local performance by passenger affluence (Figures 5–8). These figures generally show that the improvement in the dwell time modeling from RF–RO to RF–All, i.e., when taking PF and M variables into account, lies in situations with a high passenger affluence. These account for a limited share of the situations considered: around 5 to 7% of them. Yet, these are exactly the situations where the modeling of dwell time is challenging, as can be seen from the relatively large average errors made by the reference model RF–RO. In particular, the left graph of Figure 8 shows that RF–All enjoys a more steady performance, while the one of RF–RO worsens as passenger affluence increases.

Comments on local performance by deviation to scheduled arrival time (Figure 9). Figure 9 depicts how modeling errors vary with the deviation Δa to scheduled arrival time. Errors for both RF–RO and RF–All methods follow U-shaped curves with a minimum reached at $\Delta a = 0$, i.e., when trains are perfectly on time. On the first part of the U-shaped curves, i.e., for early trains, the performance of RF–RO and RF–All is virtually indistinguishable. This is certainly explained by the fact that early trains wait longer than needed in a station; therefore, passenger flows do not constrain

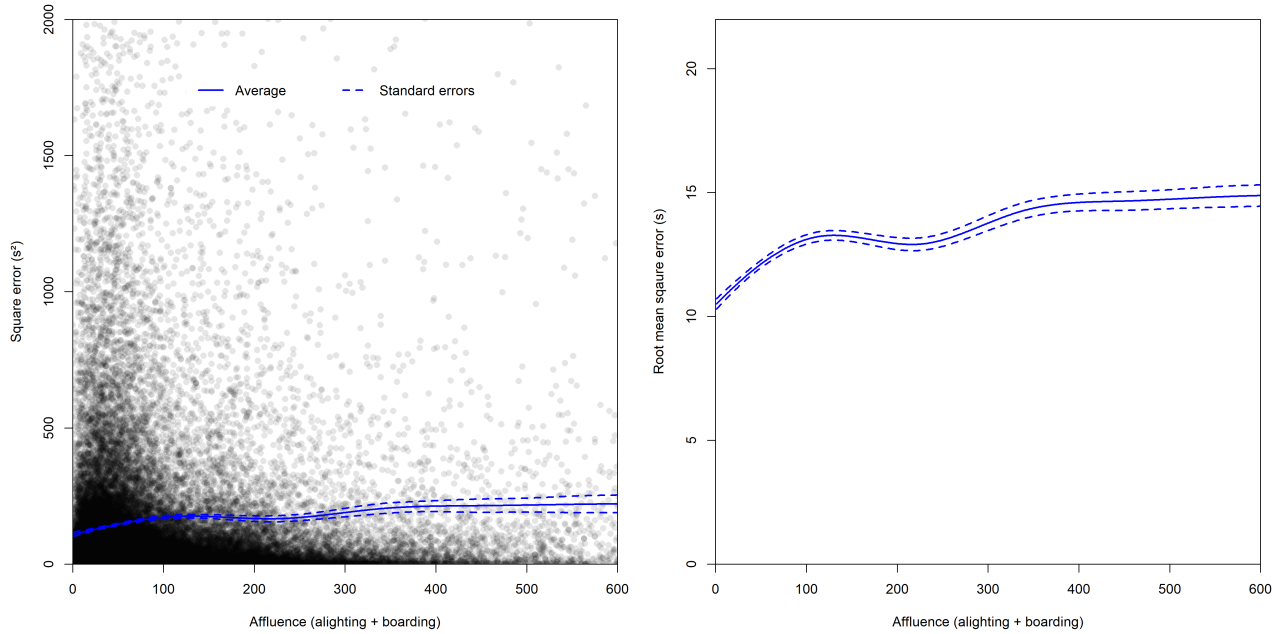


Figure 7: Left graph: scatterplot of the squared errors on the test set for dwell time modeling by RF–All plotted against passenger affluence, together with an estimation of the associated average squared errors (solid line), and standard errors thereof (dotted lines). Right graph: root of the curves obtained in the left graph, corresponding to root mean square errors by passenger affluence.

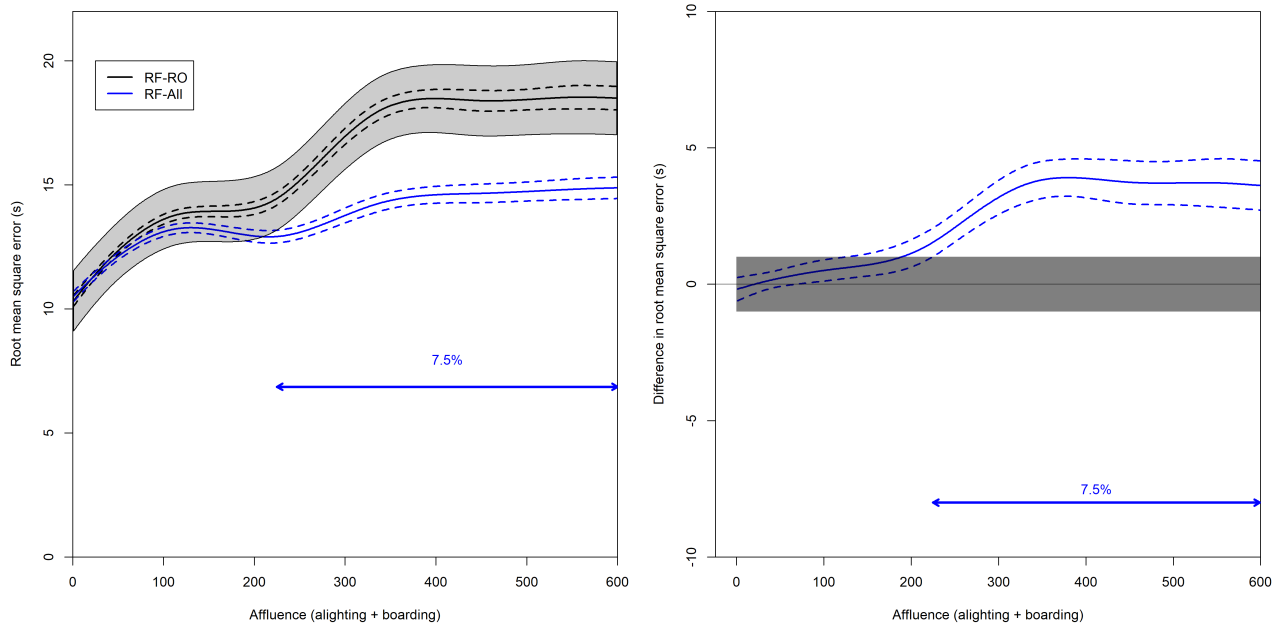


Figure 8: Left graph: root mean square errors for the modeling of dwell time (y–axis) by passenger affluence (x–axis) for RF–RO (black) and RF–All (blue). Right graph: difference of these root mean square errors, between RF–RO and RF–All. The horizontal arrows indicate the data ranges where significant improvements are achieved on average by RF–All over RF–RO; the percentages below the arrows are the corresponding data shares.

Table 8: Breakdown of the data set for line L by regimes of punctuality or passenger affluence.

Punctuality	Early: 54,697	On time: 34,388	Late: 28,242
Passenger affluence	Low: 58,813	High: 58,514	

dwell times. On the contrary, on the second part of the U-shaped curves, i.e., for late trains, RF–All consistently outperforms RF–RO, in a statistically significant manner as soon as Δa is larger than something of the order of 70 s, which accounts for 14% of the total observations in MAE (and 7% in RMSE). The improvement in performance is of order 2 – 3 s, for errors of the order of 10 – 15 s. These observations thus show that for delayed trains, passenger flows are a key determinant of dwell time, as intuition commands: trains attempt to leave a station as fast as possible when they are late on schedule.

4.4. Breakdown of the performance by regimes of punctuality or passenger affluence

The local performance study above highlights the situations where taking passenger flows into consideration helps (i.e., where RF–All is superior to RF–RO): in case of large delays to scheduled arrival time or high passenger affluence. We now clarify further these determinants by looking at their joint influence: we break down the performance by regimes of punctuality or passenger affluence.

The regimes of punctuality considered were already explained in Table 3: trains may be early, on time, or late. We define two regimes of passenger affluence, high and low, setting as threshold a quasi-median of the passenger affluences $A_{k,s,d} + B_{k,s,d}$ observed on the test data set: we use 51–52 as thresholds (low passenger affluence is passenger affluence $\leqslant 51$ and high passenger affluence is passenger affluence $\geqslant 52$). All in all, the 117,327 triplets (k, s, d) of the test data set may be broken down as indicated in Table 8.

In this section, we follow somewhat the structure of the previous analysis and first report global numerical results factored by regimes of punctuality or regimes of passenger affluence (a table), and second, provide a more local, graphical, idea of the improvement in performance of RF–All over RF–RO factored by regimes of punctuality or passenger affluence.

Global performance by regimes of punctuality or regimes of passenger affluence. Table 9 deepens the results of the fourth line of Table 7, which was devoted to random forests: the first line of Table 9 is a mere copy of the fourth line of Table 7. We then break down the performance achieved on the test data set by regimes of punctuality, i.e., compute the errors only over early trains, trains on time, or late trains. The first line of Table 9 is therefore a weighted average of its second, third, and fourth lines. We finally break down the global performance by regimes of passenger affluence.

We first comment the influence of the regime of punctuality on performance. The smallest errors are always observed for trains on time, then for early trains, while the largest errors are suffered for late trains. The influence of the subsets of variables considered is similar to what was observed already in Table 7: RO variables in isolation are more useful that PF variables in isolation, while the simultaneous consideration of RO and PF variables is even better, with the consideration of the critical door data (subset M) not changing substantially the global performance.

For regimes of passenger affluence, similar observations may be issued concerning the subsets of variables, noting however that the added value of PF variables on top of RO variables is larger in the case of a high passenger affluence than for a low passenger affluence. Generally speaking, dwell time is more difficult to predict in situations of high passenger affluence, as intuition commands.

Local performance by passenger affluence factored by regimes of punctuality (Figure 10). In Figure 10, we break down by regimes of punctuality the differences in modeling errors between

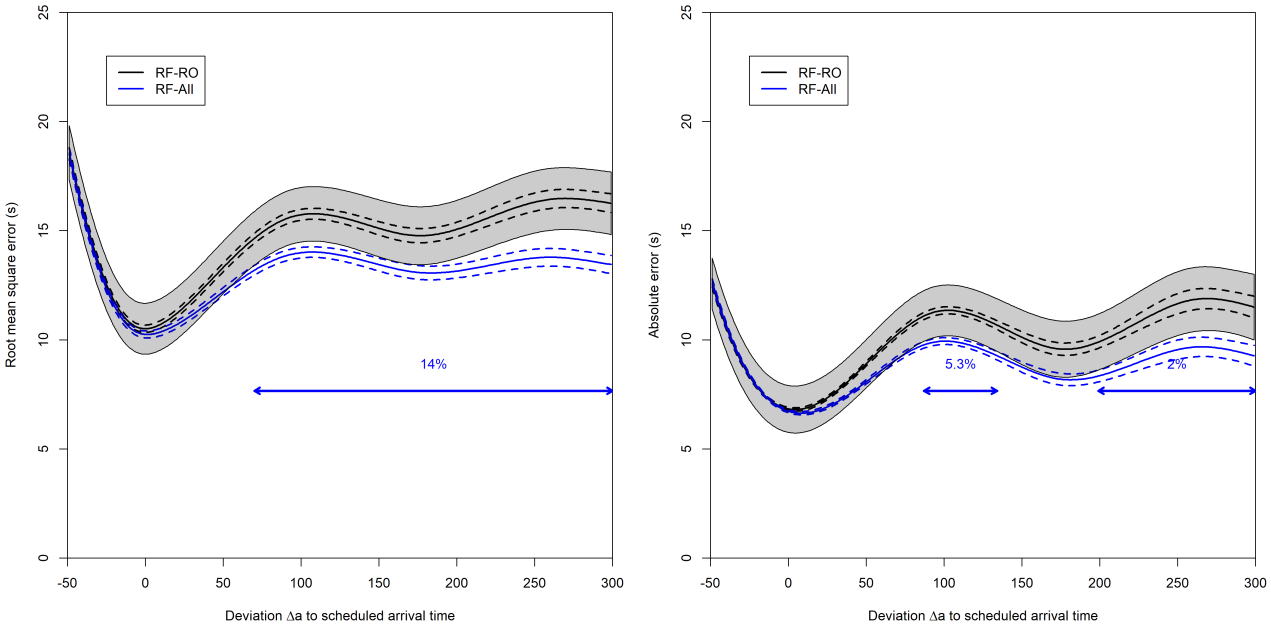


Figure 9: Root mean square errors (left graph) and absolute errors (right graph) for the modeling of dwell time (y–axis) by deviation Δa to scheduled arrival time (x–axis) for RF–RO (black) and RF–All (blue). The horizontal arrows indicate the data ranges where significant improvements are achieved on average by RF–All over RF–RO; the percentages below the arrows are the corresponding data shares.

Table 9: Modeling performance for random forests by regimes of punctuality or regimes of passenger affluence (lines) and for each subset of variables (rows); see the legend of Table 7), in MAE (*left part of the table*) and RMSE (*right part of the table*). Standard errors are smaller than 0.03 seconds.

Random forests	MAE				RMSE			
	PF	RO	RO PF	RO PF+M	PF	RO	RO PF	RO PF+M
All trains	13.7	8.4	8.1	8.0	18.8	12.9	12.5	12.3
Early trains	15.4	7.9	8.0	7.9	21.0	12.5	12.6	12.4
Trains on time	11.8	8.3	7.8	7.8	16.2	12.4	11.9	11.7
Late trains	12.7	9.7	8.5	8.5	17.2	14.2	12.9	12.7
Low passenger affluence	12.4	7.6	7.5	7.5	16.9	11.4	11.3	11.4
High passenger affluence	15.0	9.2	8.7	8.5	20.5	14.3	13.5	13.1

RF–RO and RF–ALL as functions of the passenger affluence, i.e., the graphs of Figure 10 are the counterparts of the right graphs of Figures 6 and 8. Therein, we had noticed a significant reduction of the errors in 5% to 7.5% of the cases. We observe that all these cases correspond to late trains or trains on time. (In particular, a significant reduction of the error is observed for none of the early trains.) Also, the obtained improvements are larger for late trains than for trains on time and/or take place for lower values of passenger affluence.

Local performance by deviation to scheduled arrival time factored by regimes of passenger affluence (Figure 11). We represent in Figure 11 the differences in modeling errors between RF–RO and RF–ALL as a function of the deviations Δa to scheduled arrival time, and break them down by regimes of passenger affluence. This figure is the counterpart of Figure 9, where we observed modest improvements of RF–All over RF–RO for deviations larger than something of the order of 50 s. Figure 11 shows that these modest improvements are actually associated with high passenger affluence.

4.5. Most influential variables

We recall a general methodology to determine which variables are the most influential in a random-forest modeling, explain how we implemented it on our data set, and discuss the obtained results, with a special focus on the identification of the most influential variables by regimes of punctuality.

General methodology. Two main families of methodologies exist to determine which explanatory variables are the most influential for random forests on a given data set: mean decrease accuracy [MDA] and mean decrease impurity [MDI]. Each of them may be implemented in several specific ways despite a common spirit for each methodology proposed by Breiman [2001]. We discuss below the specific implementations provided by the R package `ranger` already mentioned in Section 3.4 (see Wright and Ziegler, 2017), corresponding to the options `permutation` [MDA] and `impurity` [MDI]. The most popular criterion is probably MDA but we provide here the results obtained for both criteria.

We recall that random forests exploit instances $\boldsymbol{X}_{k,s,d}$ of vectors of variables $\boldsymbol{X} = (X_1, \ldots, X_p)$.

The spirit of MDA is the following: for each variable j, an index MDA_j is computed as follows. We first bootstrap data with replacement into T data sets (where T is the number of trees of Table 6) compute a random forest based on each of these T bootstrapped data sets, and evaluate an average difference of performance on the remainder observations of each of these data sets (the so-called out-of-bag observations): the average squared error on modified out-of-bag observations, obtained by randomly permuting the values of the variable of interest, minus the average squared error on original out-of-bag observations. The larger this average difference MDA_j, the more crucial the variable under scrutiny.

As for MDI, we recall (see Appendix A.1) that each tree $f^{(t)}$ of a forest is grown through refinements decided based on maximal reductions of in-sample errors; MDI exploits this construction: $\text{MDI}_j^{(t)}$ is simply the weighted sum of the reductions associated with the same variable j, over all refinements leading to tree $f^{(t)}$, where the weights are the proportion of observations falling in the region to be refined. The final index MDI_j is then obtained by averaging out the $\text{MDI}_j^{(t)}$ over the T trees of the forest.

In both cases, we obtain non-negative families of indices $(\text{MDA}_j)_{1\leqslant j\leqslant p}$ and $(\text{MDI}_j)_{1\leqslant j\leqslant p}$ and we depict on Figure 12 the normalized vectors

$$\frac{\text{MDA}_j}{\sum_{i=1}^{p} \text{MDA}_i} \quad \text{and} \quad \frac{\text{MDI}_j}{\sum_{i=1}^{p} \text{MDI}_i}, \qquad \text{where} \quad j = 1, \ldots, p. \tag{10}$$

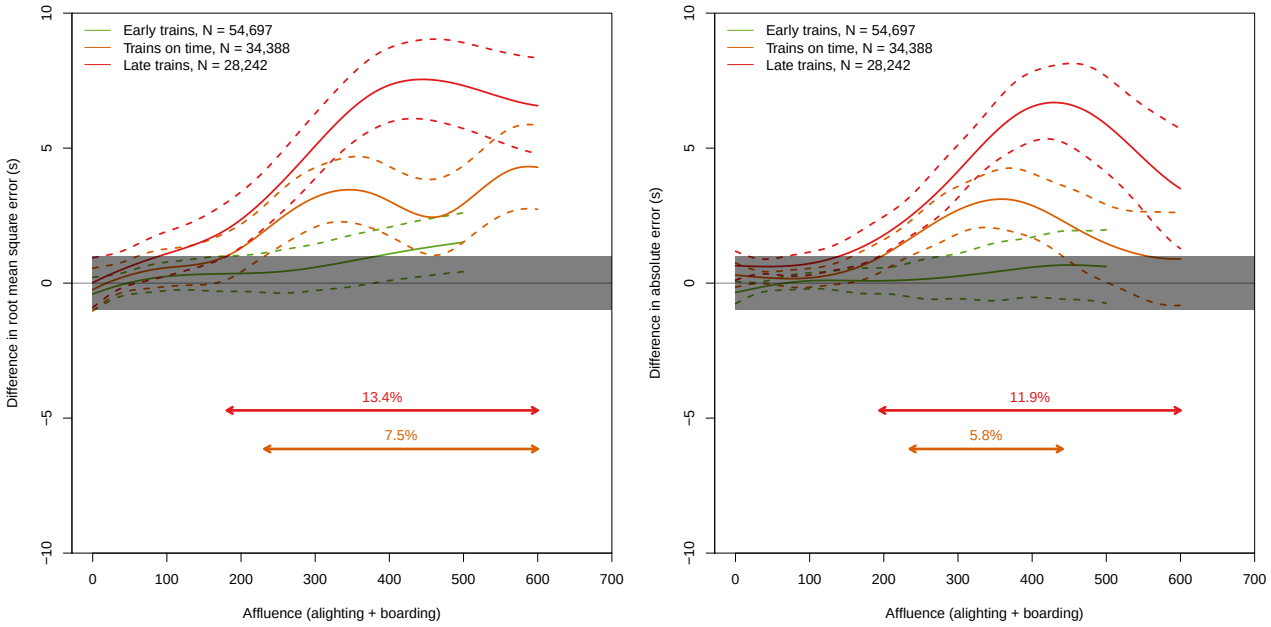


Figure 10: Differences in mean square errors (left graph) and absolute errors (right graph) between RF–RO and RF–All for the modeling of dwell time (y–axis) by passenger affluence (x–axis), factored by regimes of punctuality. Positive numbers correspond to the superiority of RF–All over RF–RO. The horizontal arrows indicate the data ranges where significant improvements are achieved on average by RF–All over RF–RO; the percentages below the arrows are the corresponding data shares.

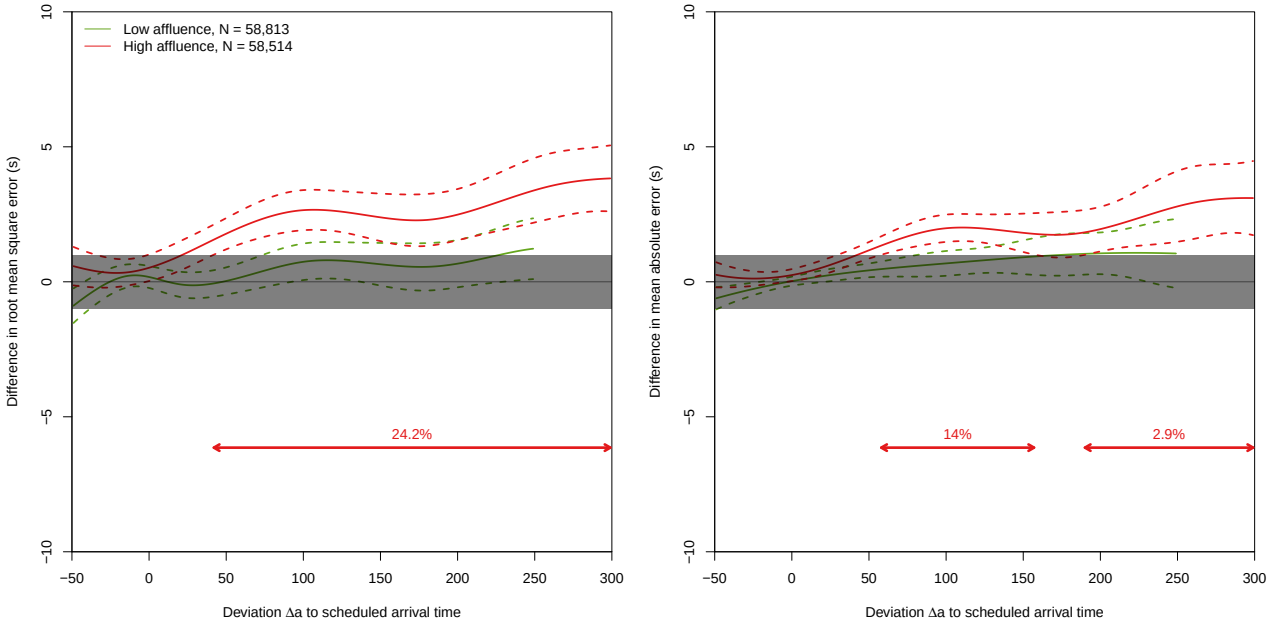


Figure 11: Differences in mean square errors (left graph) and absolute errors (right graph) for the modeling of dwell time (y–axis) by deviation Δa to scheduled arrival time (x–axis), factored by regimes of passenger affluence. Positive numbers correspond to the superiority of RF–All over RF–RO. The horizontal arrows indicate the data ranges where significant improvements are achieved on average by RF–All over RF–RO; the percentages below the arrows are the corresponding data shares.

Specific application. The first line of Figure 12 provides the normalized MDA and MDI indices for RF–All run on the entire data set; the 21 variables it relies on (see last column of Table 10) are ranked according to their normalized indices.

We also implement RF–All (still tuned with the hyperparameters of the last column of Table 6) on each of the three subsets defined by regimes of punctuality; when we do so, we only feed RF–All with 18 variables, omitting the three binary categorial variables stemming from the regime of punctuality z (given that they would be constant anyway on each of the subsets). The bottom three lines of Figure 12 provide the normalized MDA and MDI indices computed for each regime of punctuality.

Results: at a global level. Be it for MDA or MDI indices, the top three influential variables are, globally, the scheduled dwell time y^{theo}, the passenger affluence at the critical door M, and the deviation to scheduled arrival time Δa. Then come the numbers A and B of alighting and boarding passengers, as well as the crowding factor C and the fact that the train is early, i.e., $z = 1$. The importance of M may seem surprising given the modest differences in modeling performance read in Tables 7 and 9: we comment this issue in detail below.

Results: early trains. This picture for early trains is somewhat similar to the picture at the global level, except that y^{theo} and Δa have an importance much superior to other variables: they gather about 60% of the total importance. This is likely to be due to the fact that early trains are supposed to depart at the scheduled time, i.e., as mentioned earlier in Section 3.1, after a dwell time equal to $y^{\text{theo}} - \Delta a$. (We recall that $\Delta a < 0$ for early trains.) Thus, we expect that the observed dwell time y^{obs} is close to $y^{\text{theo}} - \Delta a$. Machine-learning techniques like random forests spot this kind of rules in some automatic way, which explains why y^{theo} and Δa are the most two influential variables for early trains. We remind, however, that they do so while providing a single model for all stations, all working days, all hours, and all trains (as was the title of Section 3.2).

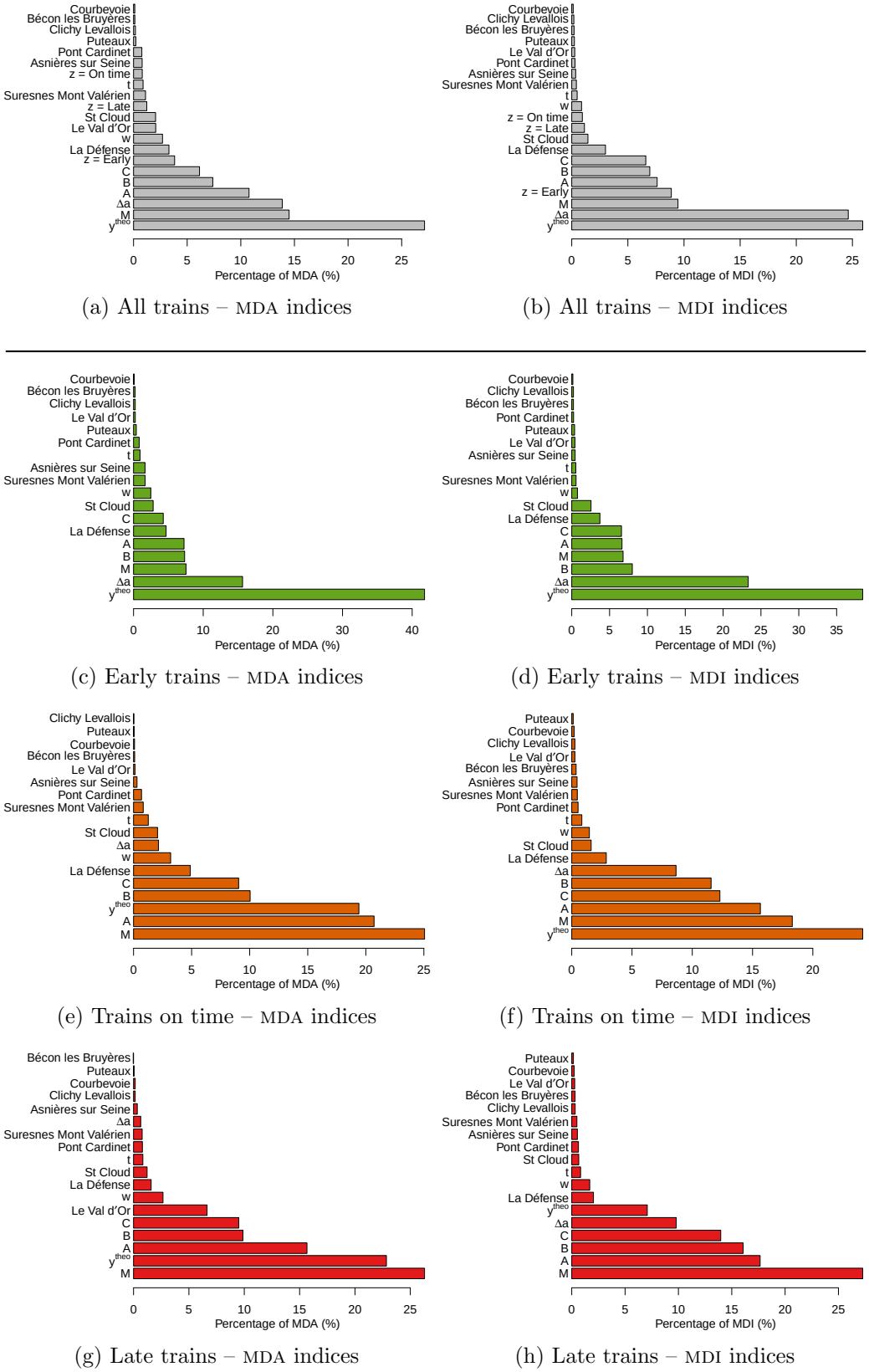
Results: trains on time and late trains. For trains on time and late trains, the MDA procedure rather points to the critical passenger affluence M as the main driving factor, with alighting number A and scheduled dwell time y^{theo} as the next most important variables. As mentioned above, this may seem surprising given the numerical results, where modest overall improvements of about 0.1 s to 0.2 s are achieved with the addition of the M variable. These modest overall improvements however hide (again) significant local improvements in critical situations, most of them related to trains on time and late trains: we noted, when producing the various graphs of Sections 4.3 and 4.4, that they reported fewer significant improvements in terms of shares of data points concerned when the variable M was omitted, i.e., when RF–[RO+PF], instead of RF–All, was compared to RF–RO. We do not provide further details for the sake of conciseness but wanted to mention this fact, as it explains the "qualitative" importance of M, which the MDA procedure confirms.

Results: stations. In all cases, stations are among the least influential variables, except maybe for La Défense and Saint-Cloud.

5. Conclusions and research perspectives

This article considers a particularly rich data set containing both railway operations variables and door-by-door passenger flows variables.

Conclusions. The main findings of our study are the following ones; they hold for the considered data set of line L and are globally robust with respect to variations on the methodology or of the considered railway line (see Appendix C).



(a) All trains – MDA indices

(b) All trains – MDI indices

(c) Early trains – MDA indices

(d) Early trains – MDI indices

(e) Trains on time – MDA indices

(f) Trains on time – MDI indices

(g) Late trains – MDA indices

(h) Late trains – MDI indices

Figure 12: The most influential variables using normalized MDA indices (left) and normalized MDI indices (right) for RF–All on the entire data set (first line) or RF–All run on sub-data sets corresponding to the regimes of punctuality (last three lines).

1. Railway operations variables are key for low modeling errors. This being said, on average and at a global level, the consideration of passenger flows variables on top of railway operations variables (only) decreases by about 0.5 s the modeling error on the observed dwell time based on mere railway operations variables.

2. However, the consideration of these passenger flows variables locally improves this modeling error in critical situations (while never deteriorating performance in non-critical situations) by sometimes up to 5 – 10 s on average: most notably, for late arrivals or for dense situations (when passenger affluence is large).

3. More generally, on this data set, railway operations variables are the most influential variables for early trains (which are constrained by the scheduled departure time and must wait possibly for an extended amount of time) while passenger flows variables are the most influential variables for late trains (which leave the station right after the passenger exchange time), and also, for trains on time. These phenomena were expected, of course, but are confirmed on data.

4. Method-wise, we discussed fully automated model-building techniques (in particular, thanks to setting their hyperparameters on data). Among them, we favored random forests but note that a closed-form linear regression model with multiplicative effects (by stations, ways and regimes of punctuality—trains that are early, on time, or late), that is also fully data driven, obtains a global modeling performance that is only slightly worse.

Discussion: alternative summaries of passenger flows variables. While we could prove the existence of an added value for passenger flows variables, there is still some room for study to determine the most efficient (effective and concise) formulation for these variables. Our rich data set includes the door-by-door numbers A^i and B^i of passengers alighting and boarding, where i ranges between 1 and I. We chose not to use all these $2I$ variables but summarized them into the total numbers A and B of passengers alighting and boarding the train (i.e., we summed up the A^i and B^i over the doors i) and also considered the maximum of their sums, $M = \max\{A^i + B^i : i = 1, \dots, I\}$. That is, we summarized the $2I$ original variables into three variables A, B, and M only. We did so for the sake of interpretability of the models built.

However, the choices made to rely on three variables only were somewhat arbitrary: to the least, the impact of these choices should be explored. The main concern (mentioned by an anonymous reviewer) is how the critical door is taken into account. It seems intuitive that the impact of passenger flows is determined by the passenger exchanges at the critical door. Now, the survey by Kuipers et al. [2021] points out that, for a given door i, identical values of the sum $A^i + B^i$ will lead to different passenger exchange times. Typically, an entirely one-directional passenger flow is fastest; then come equally balanced flows, while uneven flows tends to be more turbulent and thus lead to longer exchange times. It is therefore even unclear how to define the critical door $i^\star$, and when this is achieved, it would probably be wiser not to only consider the sum $A^{i^\star} + B^{i^\star}$ but the individual variables $A^{i^\star}$ and $B^{i^\star}$ instead, hoping that the machine-learning methods would combine $A^{i^\star}$ and $B^{i^\star}$ in some nonlinear fashion if this is relevant. We leave this issue for follow-up studies.

Other research perspectives. A main research perspective is to now provide forecasts of the dwell time. The models studied in this article rely on information (passenger exchange numbers, deviation to scheduled arrival time) that are unknown in advance but could be predicted, possibly in simple ways: the deviation to the scheduled arrival time expected at future stations equals the deviation to the scheduled departure time suffered at the present station, for instance, while passenger exchange numbers could be predicted by some average values. Doing so, and using one of the models built on historical data, we would obtain real-time predictions of the dwell time at future stations, that would get updated each time the considered train leaves a station.

Other research perspectives lies in drawing conclusions on the models built in terms of designs: design of the timetables or design of the platforms of the stations. More precisely, the model could be fed with observed (joint) distributions of deviations to scheduled arrival time and passenger flows to simulate distributions of dwell times and better design the timetables through setting a careful but possibly shorter buffer time (see Figure 1), or studying the effect of adding trains during peak hours. Also, the role and importance of the critical door on dwell time could be better understood, so as to draw conclusions in terms of physical design of the platforms, if needed.

All in all, we provided methods to output modelings of the dwell time, but these models should be extended to predictive models, or should be used for simulation and design purposes.

References

Amita, Johar, Singh, Jain Sukhvir, and Kumar, Garg Pradeep. Prediction of bus travel time using artificial neural network. *International Journal for Traffic and Transport Engineering*, 5(4):410–424, 2015. doi:10.7708/ijtte.2015.5(4).06.

Breiman, Leo. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

Buchmüller, Stefan, Weidmann, Ulrich, and Nash, Andrew. Development of a dwell time calculation model for timetable planning. *WIT Transactions on The Built Environment*, 103:525–534, 2008. doi:10.2495/CR080511.

Chen, Tianqi and Guestrin, Carlos. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

Chu, Wen-jun, Zhang, Xing-chen, Chen, Jun-hua, and Xu, Bin. An ELM-based approach for estimating train dwell time in urban rail traffic. *Mathematical Problems in Engineering*, 2015:Article ID 473432, 2015. doi:10.1155/2015/473432.

Cornet, Sélim, Buisson, Christine, Ramond, François, Bouvarel, Paul, and Rodriguez, Joaquin. Methods for quantitative assessment of passenger flow influence on train dwell time in dense traffic areas. *Transportation Research Part C: Emerging Technologies*, 106:345–359, 2019. doi:10.1016/j.trc.2019.05.008.

Daamen, Winnie, Lee, Yu-chen, and Wiggenraad, Paul. Boarding and alighting experiments: Overview of setup and performance and some preliminary results. *Transportation Research Record*, 2042(1): 71–81, 2008. doi:10.3141/2042-08.

D'Acierno, Luca, Botte, Marilisa, Placido, Antonio, Caropreso, Chiara, and Montella, Bruno. Methodology for determining dwell times consistent with passenger flows in the case of metro services. *Urban Rail Transit*, 3(2):73–89, 2017. doi:10.1007/s40864-017-0062-4.

Ding, Chuan, Wang, Donggen, Ma, Xiaolei, and Li, Haiying. Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability*, 8(11): 1100, 2016. doi:10.3390/su8111100.

Dueker, Kenneth J., Kimpel, Thomas J., Strathman, James G., and Callas, Steve. Determinants of bus dwell time. *Journal of Public Transportation*, 7(1):21–40, 2004. doi:10.5038/2375-0901.7.1.2.

Friedman, Jerome H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001. doi:10.1214/aos/1013203451.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, Cambridge, 2016.

Hansen, Ingo A., Goverde, Rob M.P., and van der Meer, Dirk J. Online train delay recognition and running time prediction. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1783–1788, 2010. doi:10.1109/ITSC.2010.5625081.

Harris, Nigel G. and Anderson, Richard J. An international comparison of urban rail boarding and alighting rates. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 221(4):521–526, 2007. doi:10.1243/09544097JRRT115.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New-York, 2nd edition, 2009.

Kecman, Pavle and Goverde, Rob M.P. Predictive modelling of running and dwell times in railway traffic. *Public Transport*, 7(3):295–319, 2015. doi:10.1007/s12469-015-0106-7.

Kuipers, Ruben A., Palmqvist, Carl-William, Olsson, Nils O.E., and Hiselius, Lena Winslott. The passenger's influence on dwell times at station platforms: a literature review. *Transport Reviews*, 41(6):721–741, 2021. doi:10.1080/01441647.2021.1887960.

Lam, William H.K., Cheung, C.Y., and Poon, Y.F. A study of train dwelling time at the Hong Kong mass transit railway system. *Journal of Advanced Transportation*, 32(3):285–295, 1998.

Levinson, Herbert S. Analyzing transit travel time performance. *Transportation Research Record*, 915: 1–6, 1983.

Li, Dewei, Daamen, Winnie, and Goverde, Rob M.P. Estimation of train dwell time at short stops based on track occupation event data: a study at a Dutch railway station. *Journal of Advanced Transportation*, 50(5):877–896, 2016. doi:10.1002/atr.1380.

Li, Yaguang, Yu, Rose, Shahabi, Cyrus, and Liu, Yan. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SJiHXGWAZ`.

Lin, Tyh-ming and Wilson, Nigel H.M. Dwell time relationships for light rail systems. *Transportation Research Record*, 1361:287–295, 1992.

Medeossi, Giorgio and Nash, Andrew. Reducing delays on high-density railway lines: London–Shenfield case study. *Transportation Research Record*, 2674(7):193–205, 2020. doi:10.1177/0361198120921159.

Palmqvist, Carl-William, Tomii, Norio, and Ochiai, Yasufumi. Explaining dwell time delays with passenger counts for some commuter trains in Stockholm and Tokyo. *Journal of Rail Transport Planning & Management*, 14:100189, 2020. doi:10.1016/j.jrtpm.2020.100189.

Pedersen, Timothy, Nygreen, Thomas, and Lindfeldt, Anders. Analysis of temporal factors influencing minimum dwell time distributions. *WIT Transactions on the Built Environment*, 181:447–458, 2018. doi:10.2495/CR180401.

Pritchard, James, Sadler, Jason, Blainey, Simon, Waldock, Ian, and Austin, Jeremy. Predicting and mitigating small fluctuations in station dwell times. *Journal of Rail Transport Planning & Management*, 18:100249, 2021. doi:10.1016/j.jrtpm.2021.100249.

Puong, Andre. Dwell time model and analysis for the MBTA red line. Technical report, Massachusetts Institute of Technology Research Memo, 2000.

Wiggenraad, Paul B. L. Alighting and boarding times of passengers at Dutch railway stations. In *TRAIL Research School*, 2001.

Wirasinghe, S. Chan and Szplett, David. An investigation of passenger interchange and train standing time at LRT stations: (ii) estimation of standing time. *Journal of Advanced Transportation*, 18(1): 13–24, 1984. doi:10.1002/atr.5670180103.

Wood, Simon. *Generalized Additive Models: An Introduction with R*. CRC Press, 2006.

Wright, Marvin N. and Ziegler, Andreas. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017. doi:10.18637/jss.v077.i01.

Yaghini, Masoud, Khoshraftar, Mohammad M., and Seyedabadi, Masoud. Railway passenger train delay prediction via neural network model. *Journal of Advanced Transportation*, 47(3):355–368, 2013. doi:10.1002/atr.193.

Zhang, Yanru and Haghani, Ali. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015. doi:10.1016/j.trc.2015.02.019.

Supplementary material for

Modeling dwell time in a data-rich railway environment: with operations and passenger flows data

Rémi Coulaud – Christine Keribin – Gilles Stoltz

Outline. This supplementary material covers the following elements.

Appendix A describes in a mathematical way the machine-learning methods used in the main body of the article, namely,

- in Appendix A.1, it does so for tree methods: random forests and gradient boosting with regression trees;
- Appendix A.2 covers feed-forward neural netwoks;
- Appendix A.3 details the influence and choice of the hyperparameters (a.k.a. tuning parameters) of the stated machine-learning methods.

Appendix B provides additional local results for line L, where performance is studied by observed dwell times.

Appendix C performs two series of robustness checks for the methodology discussed in the main body of the article, namely,

- in Appendix C.1, the (lack of) impact of modeling rather the differences $y^{\mathrm{obs}} - y^{\mathrm{theo}}$ to the scheduled dwell time y^{theo} than the observed dwell time y^{obs} itself;
- in Appendix C.2, the application of our methodology to the second data set: on line H (see Figure 2).

A. Reminder on machine-learning methods and details on their implementations

This appendix covers material omitted in Section 3, namely, it provides a mathematical description of random forests and gradient boosting with regression trees (alluded at in Section 3.4 and formally described below in Appendix A.1), and does so as well (in Appendix A.2) for the feed-forward neural network of Figure 4 of Section 3.5. Finally, the end of this appendix explains (Appendix A.3) how the train data set may used both to select hyperparameters by (5-fold) cross-validation and fit the models accordingly; it does so through a fully automated procedure (in Appendix A.3.1) while noting next (in Appendix A.3.2) that these hyperparameters have anyway a somewhat marginal influence on performance.

A.1. Tree-based methods: random forests and gradient boosting with regression trees

The two machine-learning methods described in this section both rely on the concept of a regression tree, which we review first.

Concept of a regression tree. We denote by $\boldsymbol{X}_{k,s,d}$ the feature vectors, i.e., the vectors of variables available for each triplet (k, s, d). These variables were described in Section 3.1, except that we replace the non-binary categorical variables s and z, which have S and 3 modalities, by S and 3 binary variables, respectively[2]. Table 10 indicates the size of $\boldsymbol{X}_{k,s,d}$ depending on the subset of variables used, by distinguishing components that are quantitative variables and the ones that are given by binary categorical variables.

A regression tree relies on a (hierarchically organized) partition of the feature space into finitely many regions $\mathcal{R}_1, \ldots, \mathcal{R}_R$ defined by thresholds on the components of feature vectors $\boldsymbol{X}$. Indeed, the partition stems from a binary tree, where the two children of each node are defined by a threshold level on a quantitative variable, or the values 0 and 1 of a binary variable.

Example. *A toy illustration (arbitrarily picked) with a two-level hierarchy and its associated partition is provided in Figure 13: the threshold at the root node is based on the number A of passengers alighting and uses the value* 150*, and there is a second level for the left child, which is based on the number B of passengers boarding and uses the threshold value* 200*.*

A regression tree is built on train data $\boldsymbol{X}_{k,s,d}$ in a greedy manner through successive refinements of the current binary regression tree until the refinement stops, i.e., when a node is declared a leaf. The variable and associated threshold at each node are determined by considering all possible choices thereof and by picking the pair that leads to the smallest in-sample square error for the corresponding augmented regression tree. We will consider two stopping rules, both aiming to avoid over-fitting the data. The first rule is that if one of the created children node contains fewer than 5 observations, the refinement actually does not occur, and the node at hand is declared a leaf. The second rule is to construct complete binary trees of a fixed depth (where the depth of a tree is defined by the number of nodes along the longest path from the root node down to the farthest leaf), i.e., stop refining when a certain depth is reached. This concludes the description of the construction of a regression tree; we recall that it may be identified with a hierarchical partition $\mathcal{R}_1, \ldots, \mathcal{R}_R$ of the feature vectors $\boldsymbol{X}$.

[2]Doing so, we only consider additive effects, as in (3). We also tested—but do not discuss in this article—partially multiplicative effects, for instance, replacing the component $\Delta a_{k,s,d}$ of $\boldsymbol{X}_{k,s,d}$ by the three variables $\Delta a_{k,s,d}\, \mathbb{1}_{[z_{k,s,d}=z]}$, for $z \in \{1, 2, 3\}$, that were used in (4). We did not observe significant gains in performance and were not surprised: regression-tree-based methods are per se able to deal with complex interactions between features and output (dwell time).

Table 10: Size of the feature vectors $\boldsymbol{X}_{k,s,d}$ for line L (for which there are $S = 10$ stations), depending on the subsets of variables considered.

	Variables used			
Size of $\boldsymbol{X}$	PF	RO	RO+PF	RO+PF+M
Quantitative components	3	2	5	6
Binary components	11	15	15	15
Total number	14	17	20	21

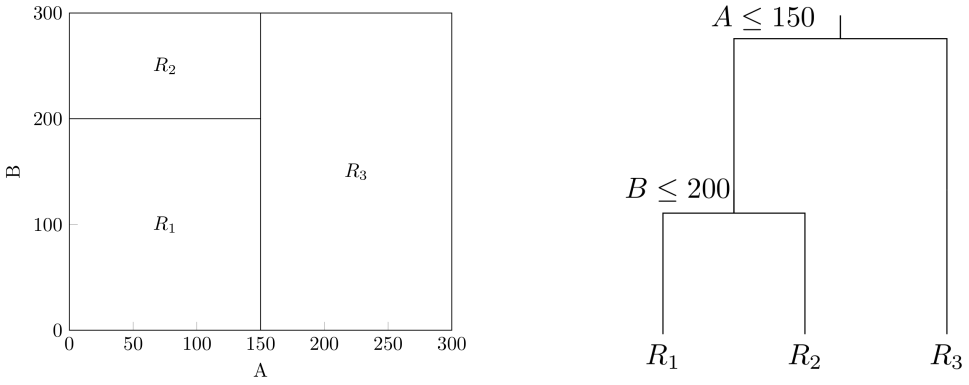


Figure 13: Toy example (arbitrarily picked) of a regression tree: the partition with 3 elements in terms of values of the variables A and B (*left*) and the associated binary tree (*right*).

Then, when a new feature vector $\boldsymbol{X}$ is to be handled, the method first identifies in which region $\mathcal{R}(\boldsymbol{X})$ of the partition $\mathcal{R}_1, \ldots, \mathcal{R}_R$ this feature vector lies. The modeled dwell time $f(\boldsymbol{X})$ for this new feature vector $\boldsymbol{X}$ finally equals the empirical average of the values $y^{\text{obs}}_{k,s,d}$ of those feature vectors $\boldsymbol{X}_{k,s,d}$ that lie in the same region $\mathcal{R}(\boldsymbol{X})$, if there is at most one such vector (otherwise, an arbitrary value is output):

$$f(\boldsymbol{X}) = \frac{1}{\sum_{k,s,d} \mathbb{1}_{[\boldsymbol{X}_{k,s,d} \in \mathcal{R}(\boldsymbol{X})]}} \sum_{k,s,d} y^{\text{obs}}_{k,s,d} \, \mathbb{1}_{[\boldsymbol{X}_{k,s,d} \in \mathcal{R}(\boldsymbol{X})]} \,. \tag{11}$$

The response function f is piecewise constant (it is constant over each member $\mathcal{R}_r$ of the partition).

One major problem of regression trees comes from their instability, which is due to their hierarchical construction: small variations in data may affect the choices made in the higher nodes and result in drastically different final results. To overcome this issue, two methods were proposed by the machine-learning literature: random forests and gradient boosting with regression trees. Both are ensemble methods using many trees of small depth to avoid over-fitting and to reduce the variances of regression trees.

Random forests. Random forests were introduced by Breiman [2001], they consist of generating (partially at random) T regression trees $f^{(1)}, \ldots, f^{(T)}$ as described above with the first stopping rule, and by resorting to the response function given by the average of these trees:

$$f(\boldsymbol{X}) = \frac{1}{T} \sum_{t=1}^{T} f^{(t)}(\boldsymbol{X}) \,. \tag{12}$$

The number T of random trees is large, and the rationale behind the average is that model errors are therefore expected to compensate each others. Kecman and Goverde [2015] was the first to use

random forests for dwell time estimation, using RO variables. We explain now how the random trees $f^{(t)}$ are generated.

Two sources of randomness are introduced to construct each given tree $f^{(t)}$: first, the data sample used to build $f^{(t)}$ is obtained by bootstrapping (i.e., by sampling with replacement into the original data), and second, to grow the tree from this bootstrapped sample, only m variables (picked at random) out of the p variables are used. These artificial sources of randomness are useful to create independence between the trees $f^{(1)}, \ldots, f^{(T)}$.

Gradient boosting with regression trees. While random forests rely on a compensation of individual errors through bagging, gradient boosting (Friedman, 2001) iteratively builds weighted sums of regression trees by focusing on the observations with the highest model errors. The regression trees successively picked for the weighted sums are thus not independent from each other.

More precisely, the basic idea of gradient tree boosting is to consider a set $\mathcal{F}$ of possible binary trees and start with an arbitrary tree $f^{(1)} \in \mathcal{F}$; in the chosen implementation, $\mathcal{F}$ is the set of all complete binary trees of depth 6. At each iteration $t \geqslant 2$, we then construct a weighted sum $f^{(t)}$ of regression trees by first considering the modeling errors

$$e_{k,s,d}^{(t-1)} = y_{k,s,d}^{\text{obs}} - f^{(t-1)}(\boldsymbol{X}_{k,s,d}) \tag{13}$$

associated with the weighted sum $f^{(t-1)}$ of the previous step, by picking the best tree $g^{(t)} \in \mathcal{F}$ to model these errors, i.e.,

$$g^{(t)} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{k,s,d} \left(e_{k,s,d}^{(t-1)} - f(\boldsymbol{X}_{k,s,d}) \right)^2, \tag{14}$$

by picking the best step size $\alpha^{(t)} \in \mathbb{R}$ to model these errors given $g^{(t)}$, i.e.,

$$\alpha^{(t)} \in \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} \sum_{k,s,d} \left(e_{k,s,d}^{(t-1)} - \alpha g^{(t)}(\boldsymbol{X}_{k,s,d}) \right)^2, \tag{15}$$

and by finally outputting

$$f^{(t)} = f^{(t-1)} + \eta\, \alpha^{(t)} g^{(t)}, \tag{16}$$

where we consider a shrinkage parameter η. The optimizations on $f \in \mathcal{F}$ and $\alpha \in \mathbb{R}$ are performed successively and not simultaneously because of computational issues. The procedure stops after T rounds and the final modeling f equals

$$f = f^{(T)} = f^{(1)} + \eta \sum_{t=2}^{T} \alpha^{(t)} g^{(t)}. \tag{17}$$

Both η and T are parameters to be set by the user.

As indicated above, these are only the high-level ideas behind the specific method used, namely, XGBoost by Chen and Guestrin [2016], which relies on two decades of advances in boosting and tree methods.

In Section 3.4, we mentioned that our implementation (and most implementations both in machine-learning competitions and in the transportation literature, see Ding et al., 2016) focus the parameters T and η and uses default parameters of the R package `xgboost` otherwise. This is because T and η hand in hand. On the one hand, large values of T and η lead to over-fitting (i.e., building a model too close to historical data with poor generalization guarantees); on the other hand, for XGBoost to "converge"—i.e., be such that $f^{(t)}$ does not change much as t approaches T—the shrinkage parameter η and T need to be small enough. All in all, a good balance between T and η should be achieved.

A.2. Feed-forward neural networks

The architecture considered for our feed-forward networks was depicted in Figure 4. It is composed of an input layer, of H hidden dense layers (each with N nodes), and of an output layer. It corresponds to inductively constructing the modeling $f(\boldsymbol{X}_{k,s,d})$ as follows. The output function $f^{(1)}$ of the first layer $h = 1$ takes a p–dimensional vector $\boldsymbol{X} = (X_1, \ldots, X_p)$ as argument, where p is provided by Table 10, and outputs a vector of length N, based on real weights $\theta_{j,n,0}$, on intercepts $b_{n,0}$, and on the so-called rectified linear unit (ReLU) activation[3] function $\sigma(x) = \max\{x, 0\}$:

$$f^{(1)}(\boldsymbol{X}) = \left(\sigma\left(b_{n,0} + \sum_{j=1}^{p} \theta_{j,n,0} X_j\right)\right)_{n\in\{1,\ldots,N\}}. \tag{18}$$

The hidden layers $h \in \{2, \ldots, H\}$ are then each associated with a function $f^{(h)}$ based on the components $f_{n'}^{(h-1)}$ of $f^{(h-1)}$, on real weights $\theta_{n',n,h}$ of the arc connecting node n' of hidden layer $h-1$ and node n of hidden layer h, on intercepts $b_{n,h}$, and on the ReLU activation function σ:

$$f^{(h)}(\boldsymbol{X}) = \left(\sigma\left(b_{n,h} + \sum_{n'=1}^{N} \theta_{n',n,h}\, f_{n'}^{(h-1)}\right)\right)_{n\in\{1,\ldots,N\}}. \tag{19}$$

The final function $f_{\boldsymbol{\theta}}$ is then based on $f^{(H)}$ and on a final series of real weights $\theta_{n,H+1}$ and on a final intercept b_{H+1} :

$$f_{\boldsymbol{\theta}} = b_{H+1} + \sum_{n=1}^{N} \theta_{n,H+1}\, f_n^{(H)}\,; \tag{20}$$

here, we collected all parameters (weights and intercepts, of all layers) into a vector denoted by $\boldsymbol{\theta}$.

The final function f is obtained by fitting $\boldsymbol{\theta}$ on data:

$$f = f_{\hat{\boldsymbol{\theta}}}\,, \qquad \text{where} \qquad \hat{\boldsymbol{\theta}} \in \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{k,s,d} \left(y_{k,s,d}^{\text{obs}} - f_{\boldsymbol{\theta}}(\boldsymbol{X}_{k,s,d})\right)^2. \tag{21}$$

Efficient gradient-descent techniques (the so-called gradient back-propagation algorithm) exist to perform the optimization leading to the value of $\hat{\boldsymbol{\theta}}$ (which is called "training the network") and they are included in the R package `tensorflow` used in our implementation.

A.3. Details on hyperparameters (a.k.a. tuning parameters) of these machine-learning methods

Section 3.6 briefly explains that machine-learning methods need to pick some hyperparameters on the train data set—two per method, which we indicated in Table 5—and fit a model on the same train data set based on these hyperparameters. In this appendix, we describe in detail the cross-validation methodology we used to perform this selection (Section A.3.1). We then illustrate through a sensitivity analysis (Section A.3.2) that the selection of these hyperparameters is not crucial for the specific data set used.

A.3.1. Automatic selection of tuning parameters through 5-fold cross-validation

Even if the performance is not (much) sensitive to the choice of tuning parameters as we will see in the next section, we alleviate the burden of users by providing a fully automated procedure to select

[3] Li et al. [2018] also use the ReLU activation function while Yaghini et al. [2013] and Amita et al. [2015] use instead a sigmoid activation function $x \mapsto 1/(1 + \mathrm{e}^{-x})$. We picked the ReLU activation function mostly because of its popularity, as asserted by Goodfellow et al. [2016].

these parameters. This procedure uses two passes on the train data set: in the first pass (Steps 1 and 2 of Figure 14), it selects the tuning parameters through a 5-fold cross-validation estimation of performance in generalization (more detail are provided below). In the second pass (Step 3 of Figure 14), it fits the model on the entire train data set based on the selected tuning parameters. We do so to avoid over-fitting issues on the train data set: selecting the best tuning parameters on the train data set by comparing the performance of models fit on the entire train data set is prone to biases; indeed, with such a procedure, we would be comparing some in-sample errors rather than out-of-sample errors, which is what we need. Put differently, this simpler procedure would evaluate the respective performance of tuning parameters in too optimistic a way.

The first pass is a 5-fold cross-validation estimation of performance which consists of separating the train data set in a random partition with 5 folds (i.e., in 5 random non-overlapping subsets), fitting the model on 4 of them and evaluating the obtained performance on the 5th fold. This 5th fold varies, and we average out the five measures of performance obtained to determine the best tuning parameters.

The tuning parameters selected by this two-pass procedure were provided in Table 6. There is an important variability in the specific values selected for the pairs of tuning parameters by the subsets of variables considered. However, the performance of a pair of a given cell of Table 6 (i.e., for a given pair of a method plus subset of variables) is not even 0.1 s apart from the performance obtained by the pair of another cell in the same line of the table (i.e., for the same method but for a different subset of variables). The seemingly instability of the values of the tuning parameters hides a remarkable stability in the underlying performance, which will be exhibited in the sensitivity analysis that comes next.

A.3.2. Sensitivity analysis

We illustrate in Figures 15–17 how tuning parameters affect performance, both in RMSE and MAE, when taking all variables (RO, PF and M ones) into account; similar conclusions are reached for subsets of variables. On these figures, we report the performance obtained on the test data set by fitting models on the train data set based on each pair of tuning parameters of Table 5. The overall conclusion is that many pairs of tuning parameters lead to an approximately equal performance, and that these pairs consist of large enough parameters, while some parameters that are too small may lead to suboptimal performance. We conclude that the selection of tuning parameters is not a crucial issue for the specific data set used.

More precisely, the sensitivity of random forests is illustrated in Figure 15; for clarity, we represent only a part of all possible pairs (m, T) of tuning parameters. Pairs with numbers of variables $m \geqslant 6$ and numbers of trees $T \geqslant 50$ obtain basically the same performance. The stability of performance is even more remarkable for feed-forward neural networks, as can be seen on Figure 17: all pairs exhibit a performance that lies in a range of radius of order ± 0.5 s. There is slightly more instability for gradient boosting with regression trees, even though taking a large number of trees (several hundreds) eventually equalizes all performance; see Figure 16.

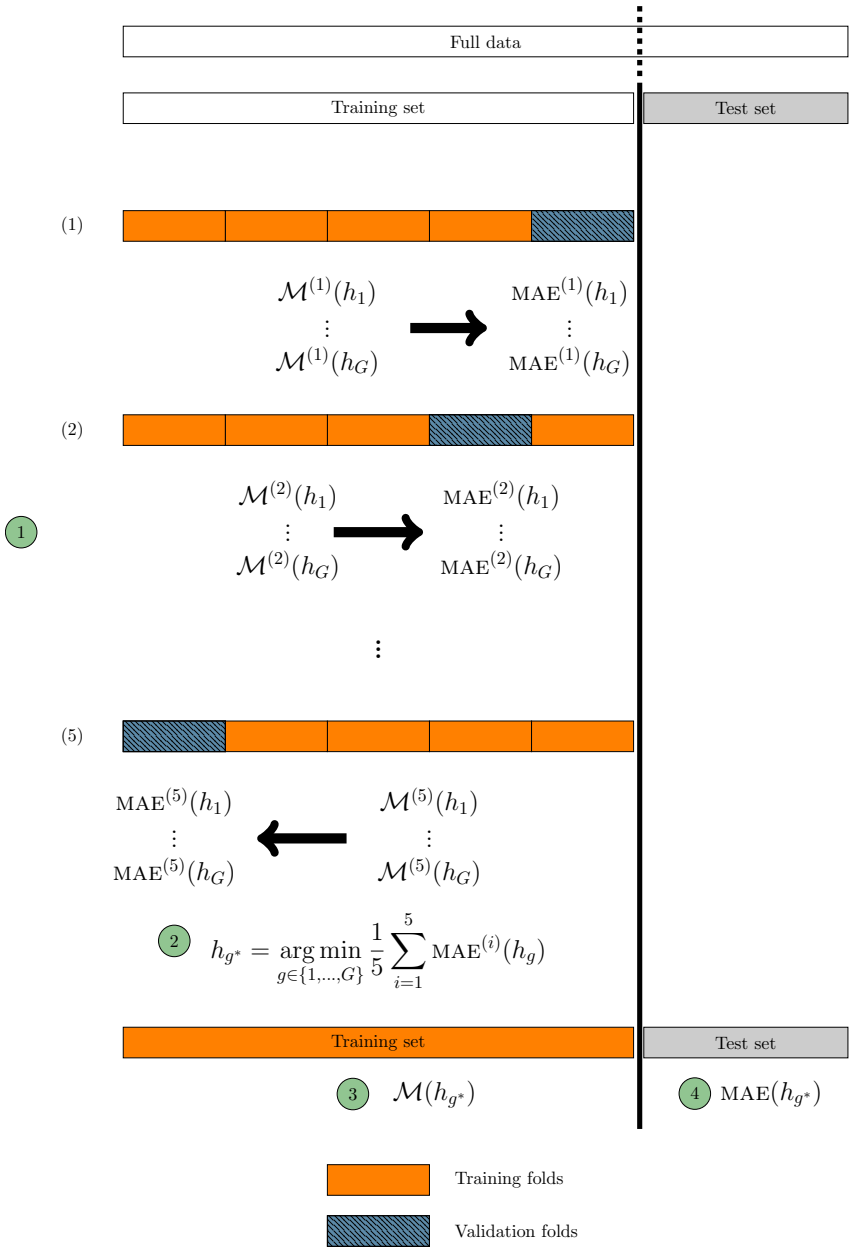


Figure 14: Principle of the automated selection procedure proposed, for a given machine-learning method. The full dataset is split into a train set (in orange) and a test set (in grey). The train set is itself split into a random partition consisting of 5 subsets called folds.
1 For each pair of hyperparameters h_g, the model $\mathcal{M}^{(i)}(h_g)$ is fit on all train data but the one of fold i and the performance $\text{MAE}^{(i)}(h_g)$ is computed for this model on fold i.
2 The performance in generalization of the method for hyperparameters h_g is estimated by averaging the five errors $\text{MAE}^{(i)}(h_g)$, for $i \in \{1, \dots, 5\}$. We then select the hyperparameters h_{g^*} minimizing $\text{MAE}(h_g)$.
3 Model $\mathcal{M}(h_{g^*})$ is fit on the entire train data set.
4 We compute and report the performance of $\text{MAE}(h_{g^*})$ of the model $\mathcal{M}(h_{g^*})$ on the test data set.

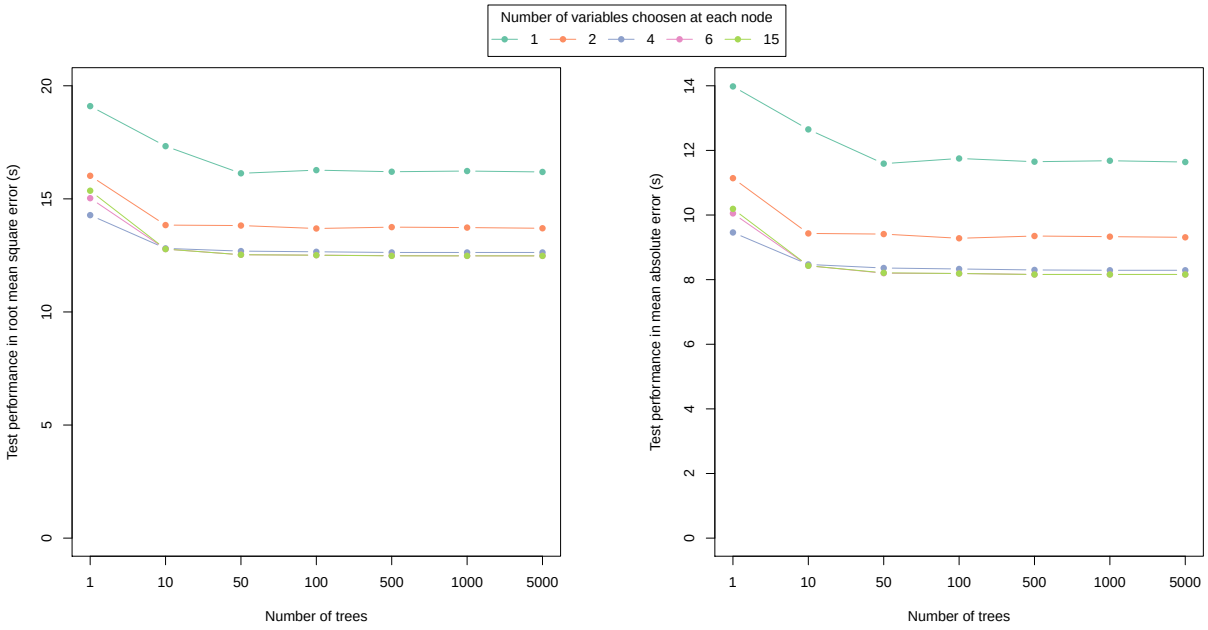


Figure 15: Performance of random forests on the test data set for various values of the pair (m, T) of tuning parameters, where m is the number of variables chosen at each split and T is the number of trees in the forest. The left picture measures performance in RMSE and the right picture does so in MAE.

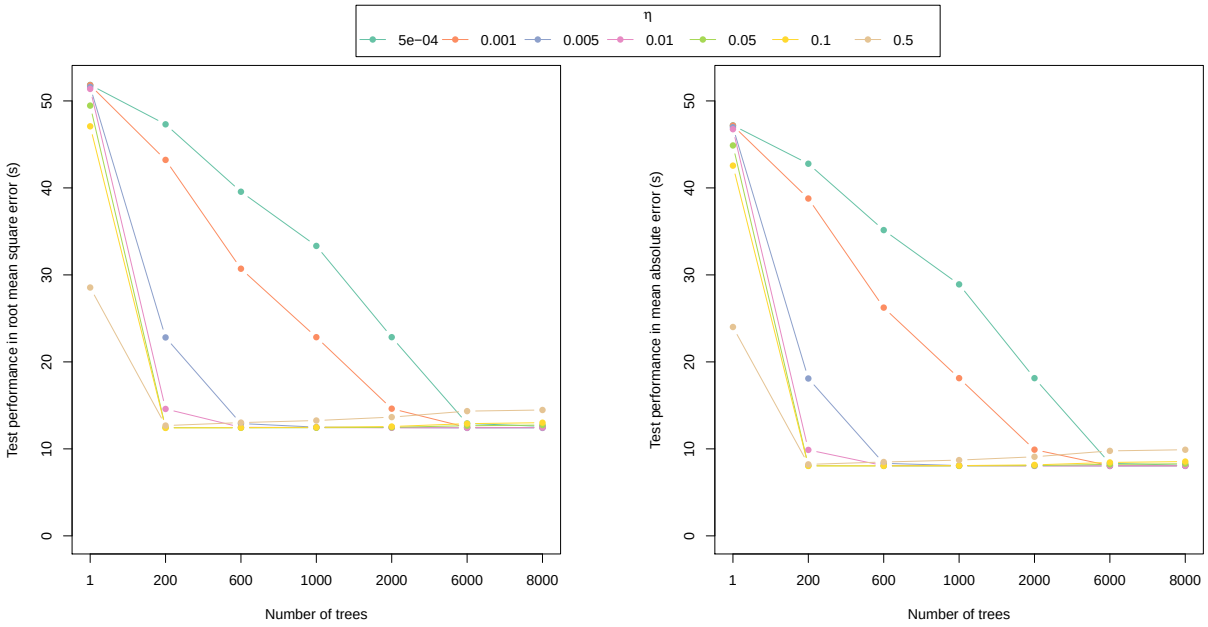


Figure 16: Performance of gradient boosting with regression trees on the test data set for various values of the pair (η, T) of tuning parameters, where η is the shrinkage parameter and T is the number of trees. The left picture measures performance in RMSE and the right picture does so in MAE.

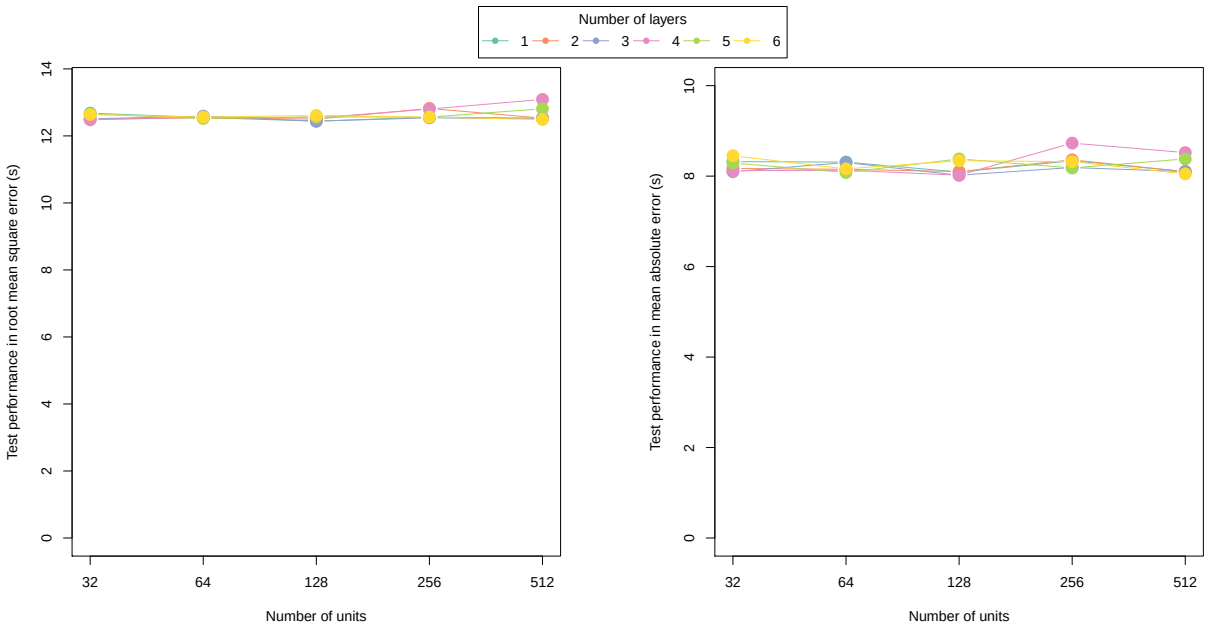


Figure 17: Performance of feed-forward neural networks on the test data set for various values of the pair (H, N) of tuning parameters, where H is the number of hidden layers and N is the number of nodes. The left picture measures performance in RMSE and the right picture does so in MAE.

B. Additional local results for line L: by observed dwell time

We provide additional results for line L, about the local performance of random forests by observed dwell time, with or without a breakdown by regimes of punctuality or passenger affluence.

With no breakdown (Figure 18). Figure 18 depicts how modeling errors vary with the observed dwell time y^{obs}. Errors for both RF–RO and RF–All methods follow U-shaped curves, with a low plateau in the 40 s – 90 s range, and with linear increases outside of this range. The RF–All method outperforms significantly the RF–RO method on average for extreme values of the observed dwell time: short dwell times (inferior to 22 s) and long dwell times (larger than 110 s). These values only account for 3 to 5% of all observations. No scheduled dwell time is shorter than 30 s, so, observed dwell times shorter than 22 s must correspond to trains with a delay and low passenger affluence, that attempt to leave the station as early as possible. This hints at the necessity of a study of local performance by deviation to scheduled arrival time, which we provide next. We have no convincing or consistent explanations for the improvements for longer dwell times.

By regimes of punctuality (Figure 19). Figure 19 is the counterpart of Figure 18, where we broke down the differences in modeling errors as functions of the observed dwell times by regimes of punctuality. In this case as well, RF–RO and RF–All obtain the same performance on early trains (which are the only ones with observed dwell times larger than 100 s). Improvements in performance are mostly due to late trains, for which between 20% and 30% of the data points are better modeled; improvements due to trains on time are negligible. These improvements take place, as in Figure 18, in a U-shaped fashion, for small and large values of the observed dwell times.

By regimes of passenger affluence (Figure 20). In Figure 20 we represent the differences in modeling errors between RF–RO and RF–All as functions of the observed dwell time, with a breakdown by regimes of passenger affluence. As in Figures 18 and 19 we observe improvements of RF–All over RF–RO for short (inferior to 20 s) or long (larger than 90 s) dwell times. But interestingly, there is a clear association of regimes: improvements for short dwell times only take place in the case of low passenger affluence, while improvements for long dwell times happen only in the case of high passenger affluence.

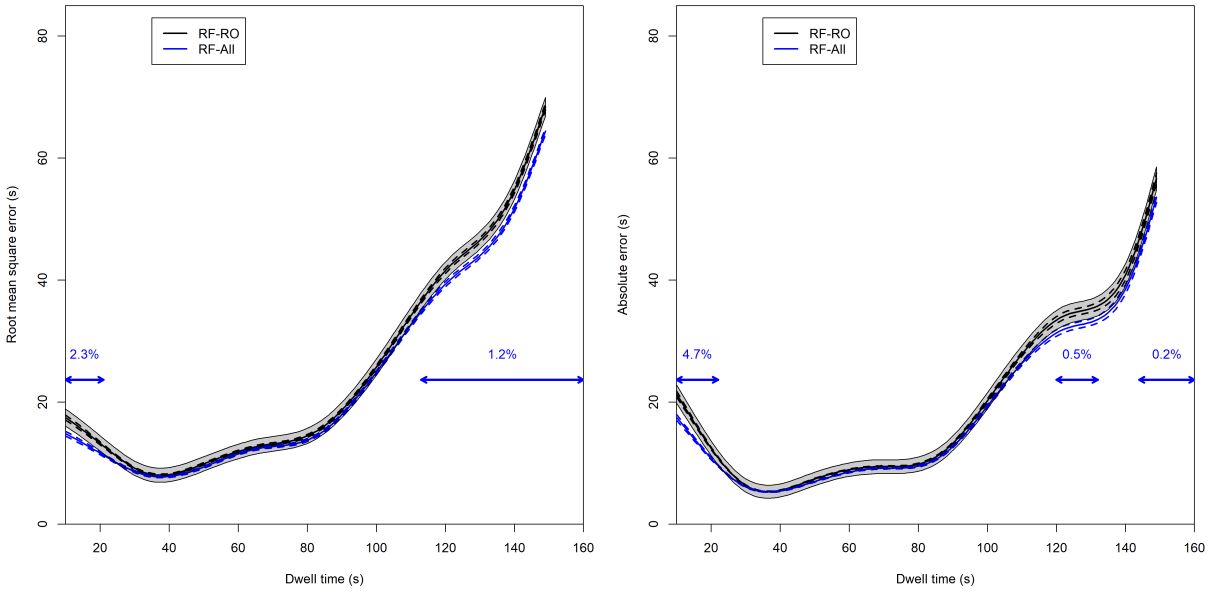


Figure 18: Root mean square errors (left graph) and absolute errors (right graph) for the modeling of dwell time (y–axis) by observed dwell time (x–axis) for RF–RO (black) and RF–All (blue).

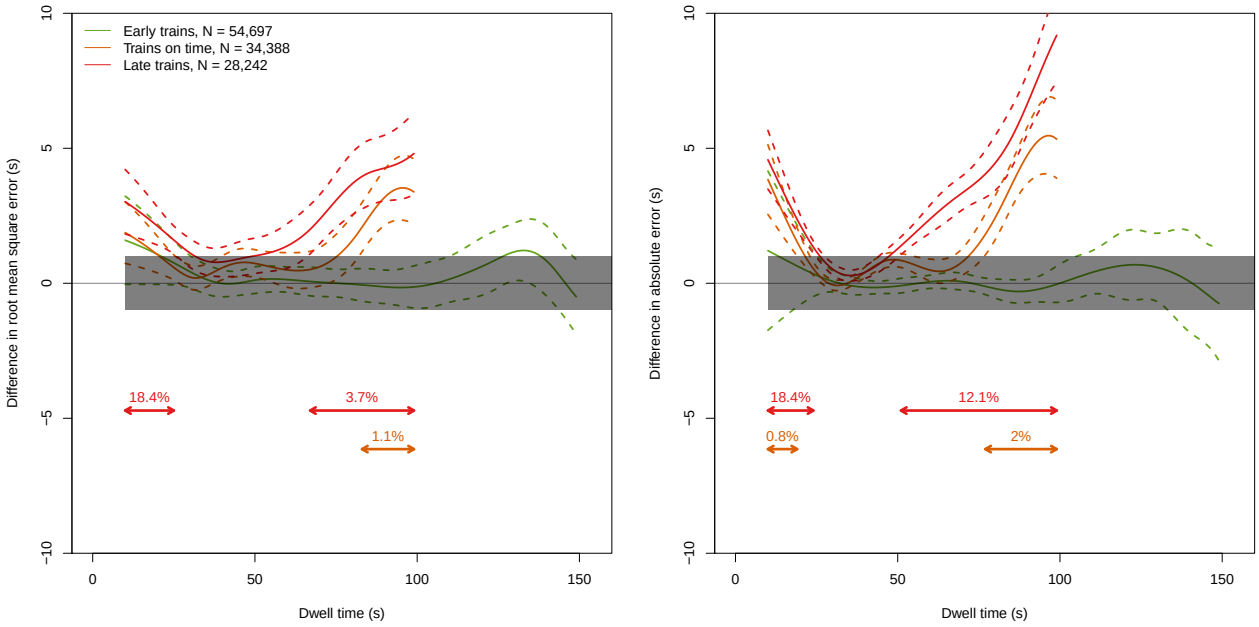


Figure 19: Differences in mean square errors (left graph) and absolute errors (right graph) between RF–RO and RF–All for the modeling of dwell time (y–axis) by observed dwell time (x–axis), factored by regimes of punctuality. Positive numbers correspond to the superiority of RF–All over RF–RO.

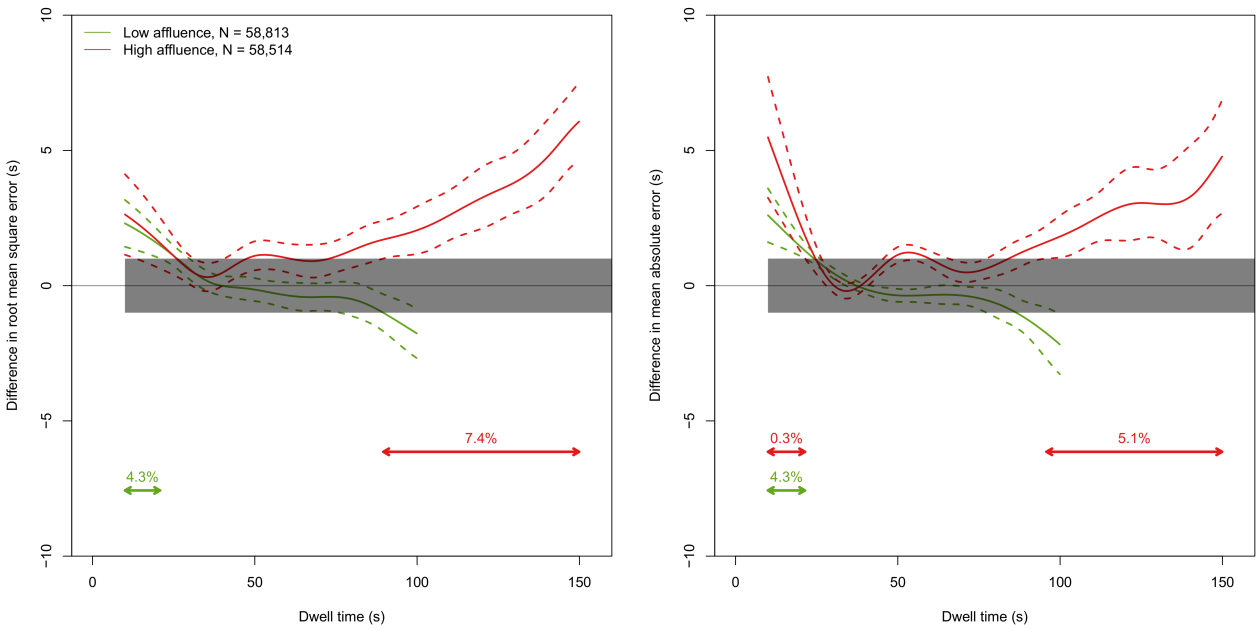


Figure 20: Differences in mean square errors (left graph) and absolute errors (right graph) for the modeling of dwell time (y–axis) by observed dwell time (x–axis), factored by regimes of passenger affluence. Positive numbers correspond to the superiority of RF–All over RF–RO.

C. Robustness checks

In this section, we discuss the robustness of our results: when the differences $y^{\text{obs}} - y^{\text{theo}}$ to the scheduled dwell time are modeled in lieu of the observed dwell times y^{obs}, still on line L (Section C.1); and what happens for the other line considered, line H (Section C.2).

C.1. Modeling the differences $y^{\text{obs}} - y^{\text{theo}}$ to the scheduled dwell time

So far, we modeled directly the observed dwell time y^{obs}. We now look instead into the modeling of the deviations $\Delta y = y^{\text{obs}} - y^{\text{theo}}$ to the scheduled dwell time y^{theo}, i.e., we run the methods discussed in Section 3 to model Δy (with newly optimized hyperparameters determined by following the methodology described in Section 3.6 and Appendix A.3.1). We obtain modelings $\widehat{\Delta y}$ which we turn into modelings $\hat{y}^{\text{obs}}$ of the observed dwell time by adding y^{theo}.

We observe in Table 11 (absolute performance of this modeling of the deviations) and in Table 12 (difference in modeling performance between direct modeling of y^{obs} and modeling of the deviations Δy) that the two approaches yield extremely similar results, except when run on the PF variables in isolation, where the modeling of deviations is significantly more efficient. About 1 s of reduction in average modeling errors is gained. This was expected as modeling Δy amounts to using the RO variable y^{theo}, which we identified as a key determinant of the observed dwell time (see, e.g., Section 4.5).

In the main body of the article, we rather performed a direct modeling of y^{obs} to be able to report some "pure" performance for the PF variables.

C.2. Results for line H—in brief

The main body of this article considered a specific sub-branch of line L (see Figure 2), and we now study what happens for the sub-branch of line H for which data is also available (see also Figure 2). This sub-branch features 11 stations on top of the origin and terminus stations. The same time period is considered as for line L and 145,609 triplets (k, s, d) are available. We may define similarly regimes of punctuality and regimes of passenger affluence (based on thresholds $\leqslant 53$ and $\geqslant 54$ passengers), and obtain the breakdown summarized in Table 13.

We (re-)optimized the hyperparameters of random forests on this new data set, following the methodology described in Section 3.6 and Appendix A.3.1, and provide in Table 14 (the counterpart of Table 9) the modeling performance of the dwell time for random forests, globally or by the regimes of punctuality or passenger affluence.

Table 14 confirms that railway operations [RO] variables are more useful for the modeling than passenger flow [PF] variables, with a difference in performance of about 2 s. This was expected. The true confirmation expected was on the improvement of RO and PF variables considered simultaneously (possibly together with the passenger affluence M at the critical door) over RO variables used in isolation: we get a mixed picture of virtually no improvement in most situations, except for late trains and in case of a high passenger affluence, where improvements of about 0.3 s are obtained. The patterns observed are therefore similar to the ones of Table 9, but take place with a lower intensity.

Figures 22 and 23 further illustrate these (more moderate) improvements in the modeling performance: no significant improvements are observed when observations are broken down by regimes of passenger affluence (Figure 23) while significant improvements are observed only in the case of late trains (Figure 22), with a significant worsening for a tiny fraction of the observations is simultaneously observed for early trains. All in all, the existence of improvements only for the most challenging situation of late trains and their smaller intensity may be caused by the absence of peaks of passenger affluence on line H (see Figure 21), while such peaks exist for line L.

Table 11: Modeling performance for the observed dwell time y^{obs} based on a modeling of the deviation $\Delta y = y^{\text{obs}} - y^{\text{theo}}$: results are formatted as in Tables 7 and 9, with standard errors still smaller than 0.03 seconds.

Methods	MAE				RMSE			
	PF	RO	RO PF	RO PF+M	PF	RO	RO PF	RO PF+M
1. Linear regression with additive effects	12.4	10.5	10.2	10.1	16.9	14.8	14.5	14.3
2. Linear regression with a multiplicative effect of Δa by z	12.4	9.1	8.9	8.8	16.9	13.6	13.2	13.1
3. Linear regression with multiplicative effects by triplets	12.2	8.8	8.3	8.3	16.7	13.2	12.6	12.5
4. Random forests	12.5	8.5	8.1	8.0	17.1	13.1	12.4	12.3
5. Gradient boosting with regression trees	12.0	8.5	8.0	7.9	16.5	13.0	12.4	12.2
6. Feed-forward neural networks	11.9	8.5	7.9	7.9	16.4	13.0	12.4	12.3
Random forests								
Early trains	15.4	7.9	8.0	7.9	21.0	12.5	12.6	12.4
Trains on time	11.8	8.3	7.8	7.8	16.2	12.4	11.9	11.7
Late trains	12.7	9.7	8.5	8.5	17.2	14.2	12.9	12.7
Random forests								
Low passenger affluence	12.4	7.6	7.5	7.5	16.9	11.4	11.3	11.4
High passenger affluence	15.0	9.2	8.7	8.5	20.5	14.3	13.5	13.1

Table 12: Differences in modeling performance between Table 11 (based on a modeling of the deviation $\Delta y = y^{\text{obs}} - y^{\text{theo}}$) and Tables 7 and 9 (based on a direct modeling of y^{obs}). Negative numbers indicate a more accurate modeling in Table 11.

Methods	MAE				RMSE			
	PF	RO	RO PF	RO PF+M	PF	RO	RO PF	RO PF+M
1. Linear regression with additive effects	−1.3	0.0	0.0	0.0	−1.5	0.0	0.0	0.0
2. Linear regression with a multiplicative effect of Δa by z	−1.3	0.0	0.0	0.0	−1.5	0.0	0.0	0.0
3. Linear regression with multiplicative effects by triplets	−1.1	0.0	0.0	0.0	−1.3	0.0	0.0	0.0
4. Random forests	−1.2	0.1	0.0	0.0	−1.7	0.2	−0.1	0.0
5. Gradient boosting with regression trees	−0.9	0.0	0.0	0.0	−1.4	0.0	0.0	0.0
6. Feed-forward neural networks	−0.8	0.1	−0.1	−0.1	−1.0	0.0	0.0	0.1
Random forests								
Early trains	−1.8	0.0	−0.1	0.0	−2.3	0.2	−0.1	0.0
Trains on time	−0.8	0.0	0.0	0.0	−1.0	0.1	0.0	0.0
Late trains	−0.7	0.2	0.1	0.0	−0.9	0.3	0.1	0.0
Random forests								
Low passenger affluence	−1.1	0.1	−0.1	0.0	−1.5	0.2	0.0	0.0
High passenger affluence	−1.4	0.1	0.0	0.0	−1.8	0.2	0.0	0.0

Table 13: Breakdown of the data set for line H by regimes of punctuality or passenger affluence.

Punctuality	Early: 33,448	On time: 58,755	Late: 53,406
Passenger affluence	Low: 72,795	High: 72,814	

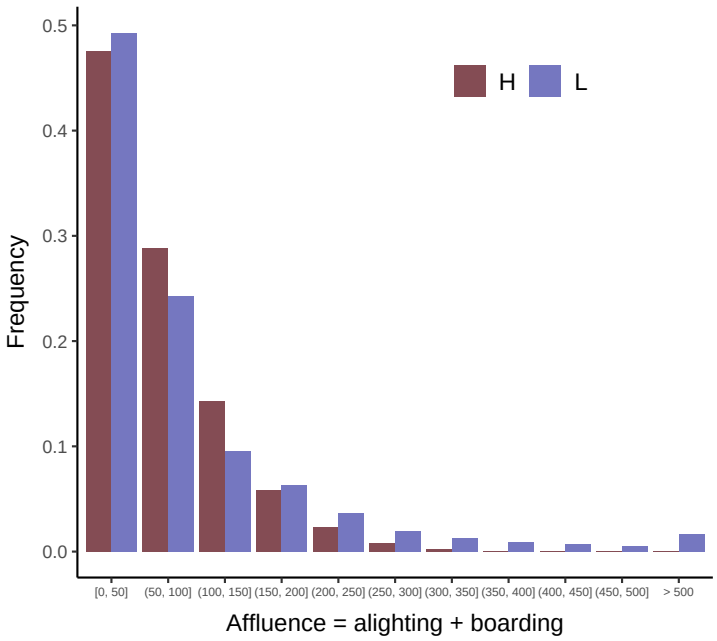


Figure 21: Histograms of passenger affluence (numbers of passengers alighting and boarding, for all stations and all trains considered) for lines H and L.

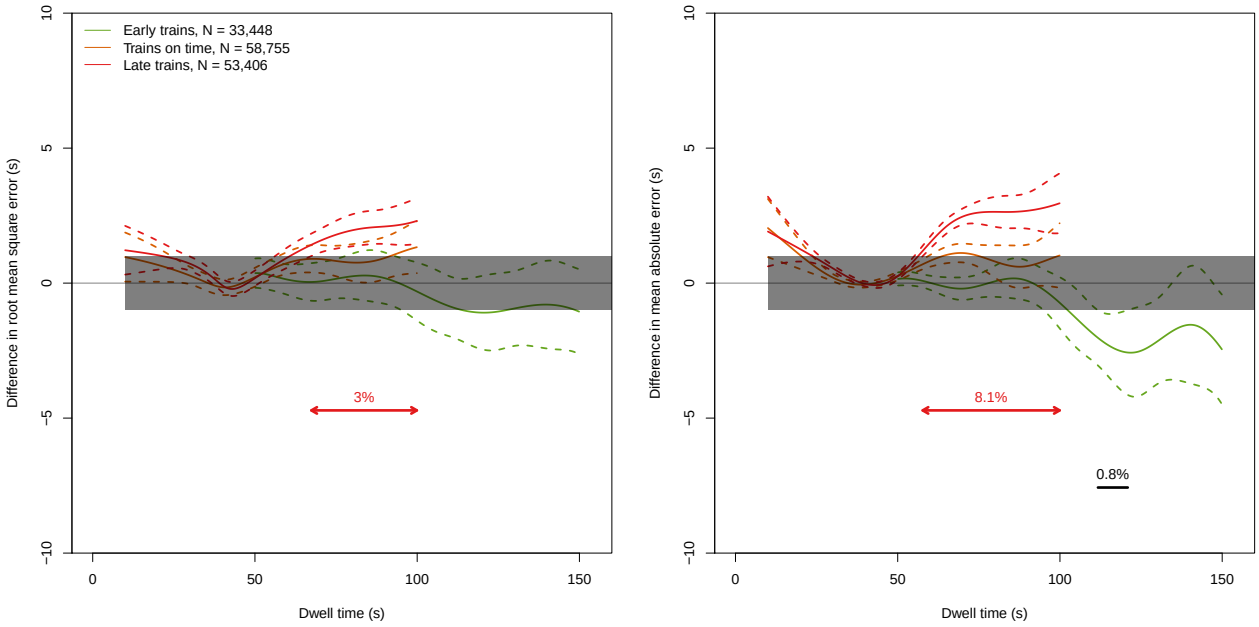


Figure 22: Differences in mean square errors (left graph) and absolute errors (right graph) between RF–RO and RF–All for the modeling of dwell time (y–axis) by observed dwell time (x–axis), factored by regimes of punctuality. Positive numbers correspond to the superiority of RF–All over RF–RO.

Table 14: Modeling performance of the dwell time for random forests by regimes of punctuality or regimes of passenger affluence (lines) and for each subset of variables (rows); see the legend of Table 7), in MAE (*left part of the table*) and RMSE (*right part of the table*). Standard errors are smaller than 0.03 seconds.

Random forests	MAE				RMSE			
	PF	RO	RO PF	RO PF+M	PF	RO	RO PF	RO PF+M
All trains	10.6	8.0	7.9	7.8	15.1	11.9	11.7	11.6
Early trains	13.2	7.8	7.9	7.9	19.2	11.9	11.9	11.9
Trains on time	9.6	7.8	7.7	7.6	13.4	11.6	11.5	11.4
Late trains	10.1	8.4	8.1	8.0	13.8	12.3	12.0	11.8
Low passenger affluence	9.9	7.5	7.4	7.4	13.7	10.9	10.8	10.8
High passenger affluence	11.3	8.5	8.4	8.2	16.4	13.0	12.6	12.5

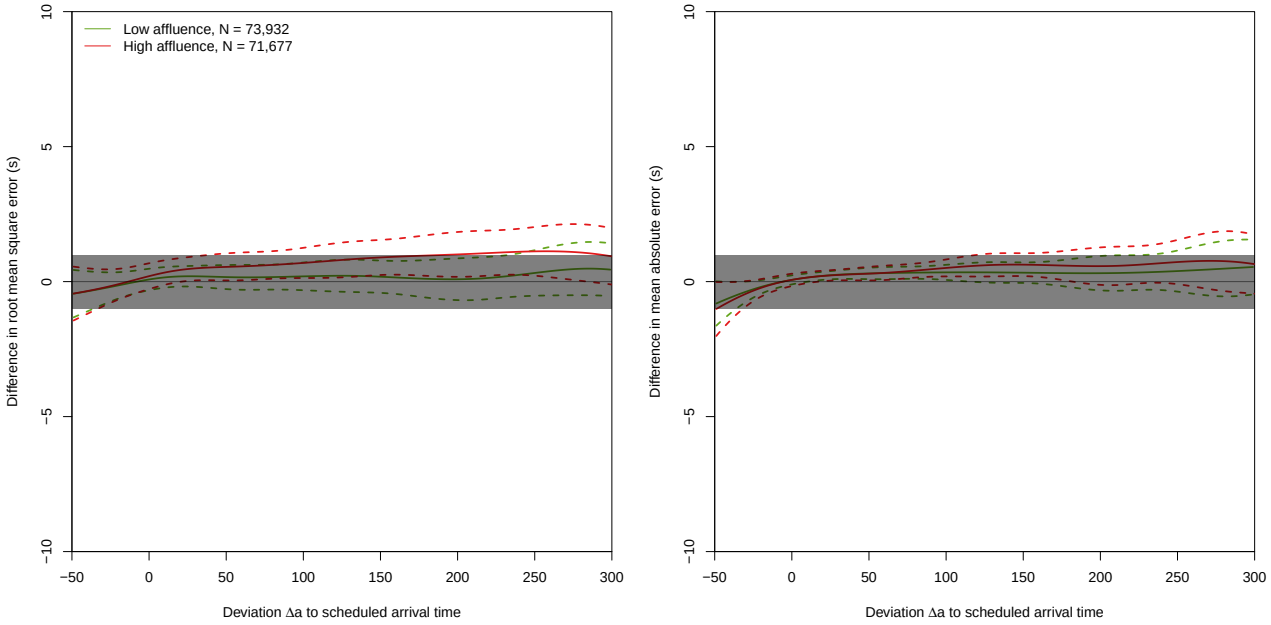


Figure 23: Differences in mean square errors (left graph) and absolute errors (right graph) between RF–RO and RF–All for the modeling of dwell time (y–axis) by deviation Δa to scheduled arrival time (x–axis), factored by regimes of passenger affluence. Positive numbers correspond to the superiority of RF–All over RF–RO.