Machine learning-assisted microscopic public transportation simulation: Two coupling strategies

Younes Delhoum, Olivier Cardin, Maroua Nouiri, Mounira Harzallah

Machine Learning-Assisted Microscopic Public Transportation Simulation: Two Coupling Strategies

Younes Delhoum[1], Olivier Cardin[1], Maroua Nouiri[1], Mounira Harzallah[1]

[1] *Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France (younes.delhoum@univ-nantes.fr; olivier.cardin@univ-nantes.fr; maroua.nouiri@univ-nantes.fr; mounira.harzallah@univ-nantes.fr)*

Abstract

Evaluating the performance of public transportation, such as bus lines for example, is a major issue for Public Transportation operators. To be able to integrate specific and local behaviors, microscopic line simulations, modelling each buses on a daily basis, provide actual added value in terms of precision and quality. Carrying out more realistic and accurate simulations requires the use of appropriate parameters. To achieve this, machine learning models trained on real-world data can be used to feed and parameterize simulation models. To address this scientific question, it is necessary to determine how to efficiently integrate machine learning and simulation models. This study aims to couple machine learning and microscopic simulation models using various strategies, evaluate their accuracy and performance and discuss the advantages and drawbacks of each. A case study involving three bus lines was conducted, with results validated against real-world data, showing a good fit for both online and offline strategies. With the best simulation time, good accuracy and adequate travel times and bus punctuality, an offline strategy seems to stand out from other coupling strategies.

Keywords: Machine learning, Microscopic simulation, ML-assisted simulation, Travel time variability, Bus travel time, Bus punctuality, Bus line simulation

1. Introduction

Rapid progress in urbanization has modernized the lives of many people, but also brought major problems and challenges, such as energy consumption, pollution and traffic congestion [1; 2]. To reduce traffic congestion, and therefore its negative impacts, cities around the world are trying to shift personal traffic to Public Transportation (PT) [3; 4; 5], also called public transport, public transit or mass transit. It is defined as passenger transportation service, provided by public or private agencies, available to the general public.

In PT, reliability is considered as one of the most critical features for evaluating Quality of Service (QoS) from the perspective of passengers and operators. Reliability refers to the certainty that passengers have about the level of service they will experience while travelling [6]. Maintaining reliable service is important for both transit passengers and transit operators [7]. Travel time, waiting time, transfer time and comfort level experienced during the transit trips are also some of the most important attributes of reliability [8], while punctuality and regularity are among the most important measures for evaluating public transport operations. Punctuality is a feature consisting of a predefined vehicle arriving, departing or passing at a predefined point at a predefined time [9].

In practice, performance indicators are often estimated purely statistically by practitioners over several months or years of data. This method provides a relatively consistent picture of the past behaviour of bus lines with low modelling effort and computational cost, but does not allow these indicators to be evaluated in different "what if?" scenarios (e.g. roadworks, new traffic priorities, etc.). To face these challenges, bus operators need efficient microscopic bus

*Corresponding author. Tel. : +33 (0)228092000
Email address: olivier.cardin@univ-nantes.fr ()

lines simulation models to evaluate the performance indicators of current lines and forecast indicators on potential developments of these lines.

Machine learning (ML), a branch of Artificial Intelligence (AI), involves using algorithms to enable systems to learn from data and make predictions or decisions, emulating aspects of human learning. ML and simulation have an area of intersection, namely machine learning-assisted simulation, describing the integration of ML into simulation [10]. Performing ML and simulation in the same framework aims at improving the quality of simulation results to some extent. However, incorporating ML techniques into the simulation model tends to make the overall framework more complex and very time-consuming.

In this paper, to couple ML models trained on actual bus data with a microscopic transit simulation model, we propose two strategies: an online strategy with more precise but time-consuming ML models, and an offline strategy that is less time-consuming but with approximated ML results. The accuracy and performance of the resulting coupled models will be evaluated using a case study based on real-world data provided by an operator. This evaluation will include an analysis of travel time day-to-day and period-to-period variability and punctuality performance indicators. In addition, a discussion on the advantages and limitations of each coupling strategy will be conducted.

The remainder of this article is structured as follows: section 2 reviews related literature, including travel time variability and prediction, traffic simulation models, as well as the coupling of ML with simulation models. Section 3 provides an overview of the proposed framework for coupling ML and microscopic simulation framework, describes and illustrates the two suggested coupling strategies in detail. A case study scenario will be described in section 4. In section 5, the simulation results will be analyzed and validated based on a set of performance indicators. A discussion of the analyses will be presented in section 6 and some concluding remarks in section 7.

2. Literature Review

2.1. Travel Time Variability and Prediction using Machine Learning

Travel time variability (TTV), also known as travel time uncertainty, is the key indicator for measuring the performance of the transport system, and can be divided into three distinct components: day-to-day variability, variability over the course of a day and vehicle-to-vehicle variability [11]. Research emphasizes day-to-day variability, which describes the degree of variation in travel time for a trip taking the same route over a specific period [12]. TTV importance can be seen from the standpoint of operators and passengers. From the operators' point of view, high TTV leads to poor performance of the transport system. This could therefore lead to a loss of passengers and thus revenue for transport operators. From a passengers' perspective, travelers expect that the transport system travel time should not exceed the average or scheduled time by more than an acceptable amount. [13].

Probabilistic distributions are capable of describing the nature and the pattern of TTV. Understanding travel time distributions and their components is a prerequisite for reliability analysis. Appropriate choice of travel time distribution is an essential element for efficient microscopic simulations of transportation and transit systems, as well as for predicting travel time and modeling discrete choices in route selections [14]. Various studies have made considerable effort to fit travel times using different types of distributions, such as **normal** [15], **lognormal** [16; 17; 18; 19; 20], **gamma** [21; 22], **Weibull** [23], **Burr** [24; 25], **Generalized extreme value (GEV) distribution** [26], **Gaussian Mixture Model** [27]. Such modelling aimed to find the best statistical distribution to describe and explain the shape and the pattern of TTV, because a distribution fitted to travel time values could illustrate a more comprehensive nature of the TTV [13].

Estimated Time of Arrival (ETA), also known as expected time of arrival, is the time at which a transportation system is expected to arrive at its destination. A number of works have been devoted to predicting ETA [28; 29; 30; 31; 32; 33; 34; 35; 36]. Moreover, ML techniques have been widely reported for travel time prediction, due to their ability to solve complex problems and extract patterns. A number of ML approaches have been widely implemented to predict the estimated bus arrival time [37; 38; 39; 40; 41; 42; 43; 44] and bus travel time [45; 46; 47; 48; 49; 50; 51; 52].

2.2. Traffic Simulation Models

Traffic simulation can be defined as the mathematical model of transportation systems, implemented through the application of dedicated computer software. They can be divided according to the level of details provided into microscopic and macroscopic traffic simulation models. Microscopic traffic simulator focuses on the movement of

each individual entity in the system. They provide a detailed picture of each individual vehicle including: location, time and speed [53]. These models are very effective in assessing heavily congested conditions. However, they are time-consuming, costly, and can be difficult to calibrate. Macroscopic traffic simulators rely on deterministic relationships between traffic flow, speed and density. They are originally developed to model traffic within specific transportation sub-networks, such as freeways and rural highways [54].

Over the past decades, a number of traffic simulation frameworks have been developed, among which:

- *TRANSIMS* developed at Los Alamos National Laboratory (USA) is an integrated set of tools for performing the analysis of a regional transportation system based on a cellular automata microscopic simulation model [55];
- MIT (Massachusetts Institute of Technology) developed *MITSIMLab*, an open-source microscopic traffic simulator that evaluates the impacts of alternative traffic management systems, public transport operations and various strategies at the operational level [56];
- *MATSim* (Multi-Agent Transport Simulation) developed by ETH Zurich, is a major open-source framework, used to implement large-scale agent-based transport simulations, it is highly modular, allowing a very high level of customization [57];
- *SUMO* (Simulation of Urban MObility) was developed at the German Aerospace Centre. It is a microscopic multi-modal traffic simulator capable of simulating different types of traffic data, in which vehicles, public transport and persons are modeled explicitly [53];
- *Starling*, an agent-based simulation software, proposed by Leblond et al.[58], was developed as a generic framework to deal with spatial issues of territories. The simulation framework is designed to evaluate a specific mobility service.

2.3. Machine Learning and Simulation Models

The fields of ML and simulation have an intersecting area, which can be divided into three subfields: 1) Simulation-assisted machine learning describes the integration of simulation into ML; 2) Machine-learning assisted simulation describes the integration of ML into simulation; and 3) A hybrid combination describes a combination of ML and simulation with a strong mutual interaction [10].

In recent years, several works have opted for simulation-assisted machine learning, including an approach using unsupervised ML algorithms to automatically detect repetitive patterns in a program's execution, to reduce costly simulation time without significant loss in accuracy [59]. Shafizadeh et al.[60] coupled ML techniques including SVM, ANNs, tree-based models and statistical models with cellular automata to simulate urban growth in the city of Tehran. Elbattah et al.[61] proposed a framework incorporating simulation modeling along with ML to design pathways and assess the return on investment of implementation. Chabanet et al.[62] combined a computationally expensive simulator with a KNN classifier, which is less computationally costly to use online but whose predictions are only approximations of the simulator.

On the other hand, ML assisted simulations have been performed in a number of studies, including a proposed simulation framework to explore the effectiveness of different ML approaches in streamflow simulation for four rivers in the United States [63]. Yan et al.[64] proposed a method for coupling neural networks and numerical models to simulate and identify high-risk areas for urban flooding, as well as to predict water accumulation depths. The outputs of the simulation models are used to feed the neural network. Shahhosseini et al.[65] proposed an approach to couple a simulation model with ML techniques, incorporating simulation results into the ML models to enhance corn yield predictions in the US Corn Belt. Abdelaty et al.[66] used the simulated energy consumption scenarios to develop four different data-driven modelling techniques.

In the field of transportation, many studies have used simulation data and results to build ML models. Al Mamlook et al.[67] evaluated and compared different ML approaches to predict road accidents, based on driving simulation data. In another work, Sroczynski et al.[68] used the simulation results to develop and test ML models for traffic prediction. ML has also been implemented to assist traffic simulation models, among which can be cited Jiang et al.[69], that used unsupervised ML techniques for parameter calibration of a traffic simulation model. Ratrout et al.[70] proposed a ML model-based calibration methodology for the PARAMICS model. Jang et al.[71] proposed a method for integrating

deep reinforcement learning into traffic simulation modeling. Daguano et al.[72] presented a method for calibrating microsimulation models using artificial neural networks.

The use of frameworks combining ML models and public transportation simulators is still relatively underutilized. Othman et al.[73] proposed a framework integrating neural network models into a public transport simulation model to improve real-time supply based on multiple demand scenarios. First, a Multi-Layer Perceptron combined with a Linear Regression model, trained on traffic data and weather information, was employed to predict congestion type, duration, and associated travel delays. These predictions are then input into a simulator to model various scenarios, with the goal of optimizing a scheduling plan in the most cost-effective manner possible.

2.4. ML-Simulation Coupling

A framework for coupling ML and simulation model will mainly include four layers: a data layer, a ML layer, a simulation layer and an evaluation layer. The coupling scheme depends on the nature of the coupling. In simulation-assisted ML approaches: input data will be used to prepare datasets to train ML models, the generated ML models will be used during the simulation, the results of which will be evaluated using key performance metrics. In ML-assisted simulation approaches, the process begins by setting up and running the simulation scenario. The results from this simulation are then used to create datasets for training ML models. These models are subsequently evaluated using appropriate performance metrics. An overview of the ML coupling scheme is shown in Figure 1.

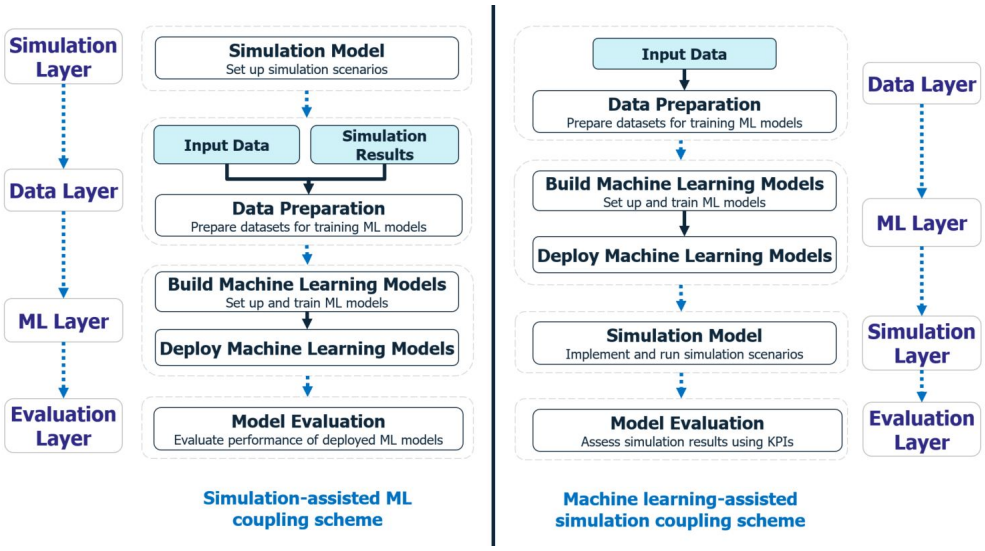

Figure 1: ML-Simulation coupling scheme

In the field of transportation, there might be a research gap related to the lack of methodology to couple ML and microscopic models for public transport simulation. Although there is limited existing works on the topic, the coupling process itself, including the exchange between ML and simulation layers, is briefly outlined in most studies. To address this research gap, this work contributes to the literature by proposing and comparing two strategies for coupling ML with a public transport simulation model. We refer by *online* the coupling strategy in which the ML models will provide PT simulation model with bus travel times, the two exchanging directly via a request-response process. On the other hand, we refer by *offline* the coupling strategy in which the simulation model will not directly exchange with the ML models, but will only have access to the resulting outputs of ML models. In other microscopic traffic simulation works, online/offline strategies refer to online/offline learning, while in this study we refer to online/offline coupling. In other words, whether a ML is accessible (online) or not (offline). In offline coupling, the results of ML are integrated (loaded) as input into the simulation model. It is also interesting to evaluate the accuracy and performance of the simulation model and to understand to what extent each of the coupling strategies can be applied. An overview of a proposed framework for coupling ML and public transport simulation model is given in Figure 2.

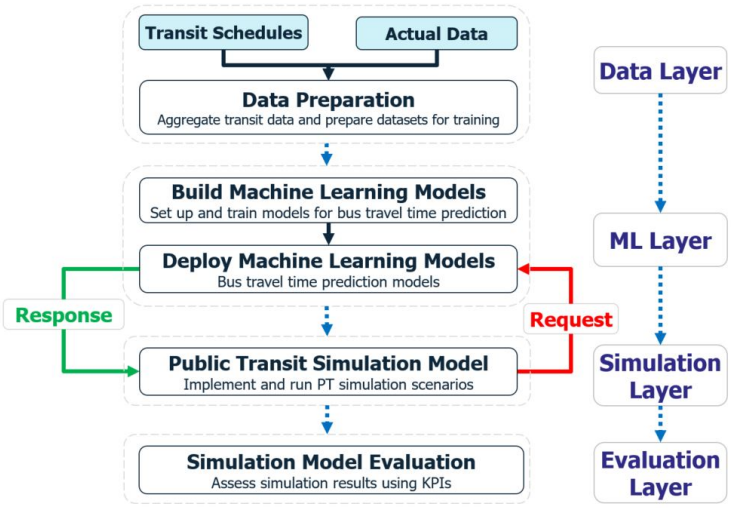

Figure 2: An overview of a framework for coupling ML and public transport simulation models. It consists of four layers: **Data layer**, in which transit schedules and actual data are used to generate aggregated datasets. **ML layer**, in which datasets will be first prepared, a ML model will be set up, then trained for bus travel time prediction, and finally deployed. **Simulation layer**, in which a simulation scenario is implemented and then simulated, in which travel times are estimated by ML models following a request-response process. **Evaluation layer**, in which the simulation will be evaluated using a set of KPIs

Reference	Simulation-Assisted ML	ML-Assisted Simulation	Transportation	Public Transport	Travel Time Prediction	Travel Time Variability	Description
Abdelaty et al.[66]	x						ML prediction models for battery-electric bus energy consumption in transit
Al Mamlook et al.[67]	x		x				ML to predict the freeway traffic accidents-based driving simulation
Chabanet et al.[62]	x	x					Coupling digital simulation and ML metamodel
Daguano et al.[72]		x	x				Automatic calibration of microscopic traffic simulation models using ANNs
Elbattah et al.[61]		x					Designing care pathways using simulation modeling and ML
Hamerly et al.[59]	x						Using ML to guide architecture simulation
Jang et al.[71]		x	x				Agent-based simulation modeling with Deep RL for smart traffic signal control
Jiang et al.[69]		x	x				Parameters calibration of traffic simulation model based on data mining
Othman et al.[73]		x	x	x	x		ML aided simulation of public transport utilization
Ratrout et al.[70]		x	x				Calibration of PARAMICS model: Application of AI-based approach
Shafizadeh et al.[60]		x					Coupling ML, tree-based and statistical models with CA to simulate urban growth
Shahhosseini et al.[65]	x						Coupling ML and crop modeling improves crop yield prediction in the US Corn Belt
Sroczynski et al.[68]	x		x				Road traffic prediction: by ML equally effectively as by complex microscopic model
Tongal et al.[63]		x					Simulation & forecasting of streamflows: ML coupled with base flow separation
Yan et al.[64]	x						A prediction model of urban flood inundation: coupling ML and numerical simulation
Proposed work		x	x	x	x	x	ML-assisted microscopic PT simulation: two coupling strategies

Table 1: Comparative table of related works

In Table 1, We compare our proposed work with existing studies based on the following aspects: whether the framework is simulation-assisted ML or ML-assisted simulation, whether the focus is on transportation and public transport, and whether the work addresses TTV and prediction.

3. Methodology

In this section, the spatial and temporal components of transit bus route, as well as the data used, will be briefly presented. The proposed approach to couple ML and simulation models, including online and offline strategies, will then be explicitly detailed and illustrated on a real case application.

3.1. Problem Statement

A transit bus travels from an origin to a destination, passing through a set of transit stops along the way. Spatial components can be grouped, according to their type. *Section* refers to the links between consecutive stops. *Segment* is made up of several consecutive sections. *Route* is formed by the join of all sections from an origin terminal to a destination terminal [74]. In order to maintain acceptable service reliability, bus operators set up control points along the bus route. Control points, also called timing points, are particular stops where bus departure times are subject to regulation or to meet a specific buffer time [8]. Additionally, timing points can be used to divide a transit route into segments. An illustration of the spatial components of bus routes is given in Figure 3.

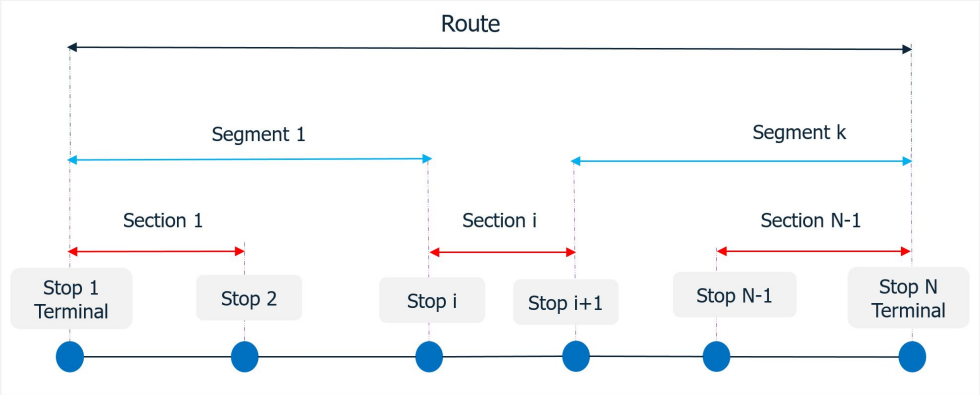

Figure 3: An illustration of the different spatial components of a transit line

The temporal components of bus routes can be divided into durations and time points. Travel time, dwell time, and running time are durations, while arrival time and departure time at bus stops are specific points in time. Travel time consists of both dwell time and running time. Dwell time refers to the period during which a bus remains stationary at a scheduled stop, while running time refers to the period during which a bus is in motion between two stops.

Depending on the level of detail, operating times can be categorized as follows:

1. **Section travel time**: This corresponds to the dwell time and running time between successive stops within a specific section.
2. **Segment travel time**: This refers to the sum of the travel times for all sections within the segment under consideration.
3. **Route travel time**: This refers to the total travel time from the first stop to the last stop.

Figure 4 provides an illustration of the temporal components of bus lines at the section level.

In this study, the data used consist of: 1) **Transit schedules**, grouped by transit line (e.g., bus line) and formatted as *GTFS* (General Transit Feed Specification) tables [75]; and 2) **Actual data**, structured similarly to a *GTFS* stop times table, providing information on transit trips during a specified period of the year, including date, vehicle ID, the corresponding trip, and actual stop times. The latter are recorded from GPS trackers on transit vehicles at each stop. Therefore, scheduled and actual stop times can be mapped in the same data structure based on mutual attributes (e.g., trip ID and stop ID).

Next, the actual observations, which are originally available at the stop level, can be transformed to the section and segment levels. This transformation aims to mitigate the impacts of imprecise stop times, which are susceptible

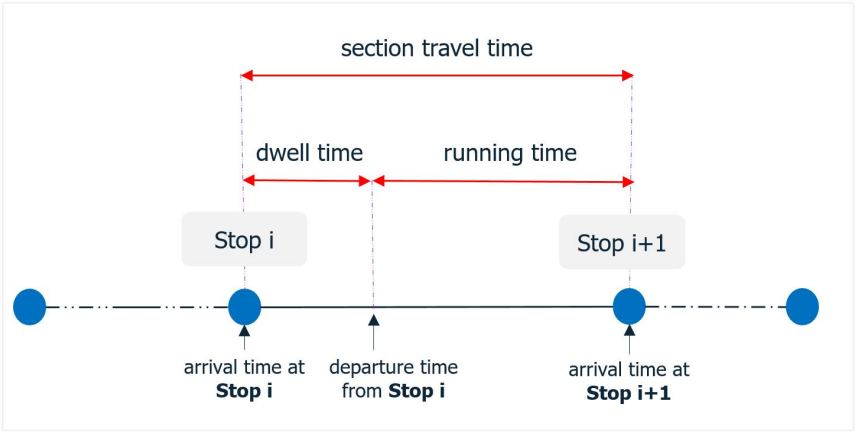

Figure 4: An illustration of the different components of travel time between two transit stops

to measurement errors that can adversely affect the quality of ML models. The components of the bus route, as well as the stop and travel times of a bus trip, aggregated at the stop, section, and segment levels, are shown in Figure 5.

Figure 5: An illustration of the components of a bus route with stop and travel times

3.2. Coupling ML and Simulation Models

Before outlining the proposed approach to couple ML and public transport simulation model within the framework, an overview of a generic traffic simulation model *Starling* [58] will be given. It combines an agent-based framework and a discrete-event approach, it is a microscopic model for mobility simulation. Its aim is to provide a basis for the development of computer models for the simulation of specific transportation systems, consisting of generic simulation classes that can be extended to match the specifications of the system being simulated.

3.2.1. Coupling Approach: Overview

The developed coupling approach was designed in order to provide public transport simulation model with more realistic travel times, based on historical data via ML models. It consists of three entities: traffic simulation model, the

deployed ML models and a ML interface which ensures the exchange between the simulator and the ML models. Additionally, in order to perform transit simulation, scenario data and operator parameters will be added to the simulator inputs. An overview of a generic simulation model coupled with deployed ML models is shown in Figure 6.

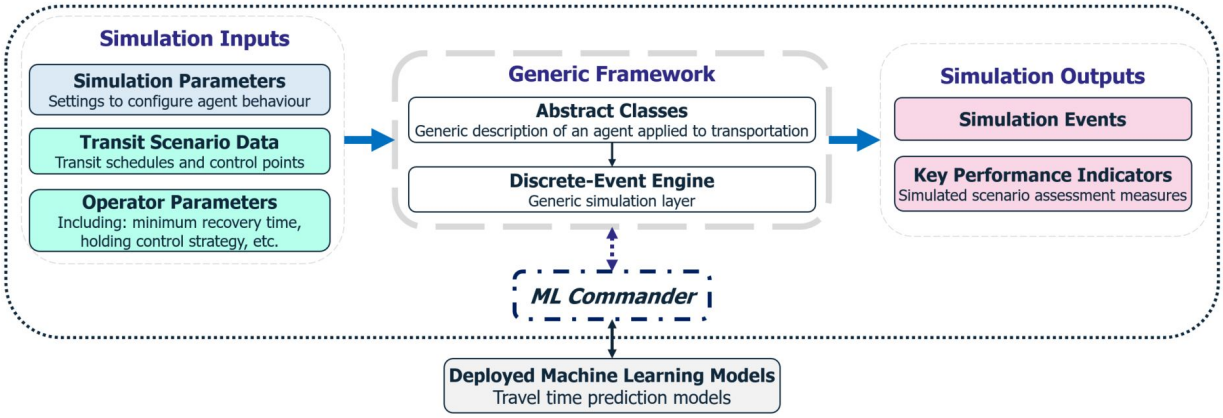

Figure 6: An overview of coupling a generic framework with ML models. It consists of two blocks: the block below groups the ML models deployed for travel times prediction; the block above refers to the simulation framework, its parameters, the scenario data as well as the operator parameters, and in addition to a ML interface (commander) which ensures the exchange between simulator and ML models

From a technical point of view, the simulator iterates through the transit trips one by one, uses the input control points to split the trip route into segments, and applies a specific process to simulate the travel segments. In essence, at each control point, the simulated bus requests its next segment travel time by providing the ML interface with the route segment and simulation information. The ML commander, on the other hand, selects the most suitable model for the requested route segment based on the request information provided. The chosen model will then predict and return the travel time of the corresponding segment based on the simulation information. Finally, the ML interface will prepare a response and send it back to the simulator. An overview of the coupling process is shown in Figure 7.

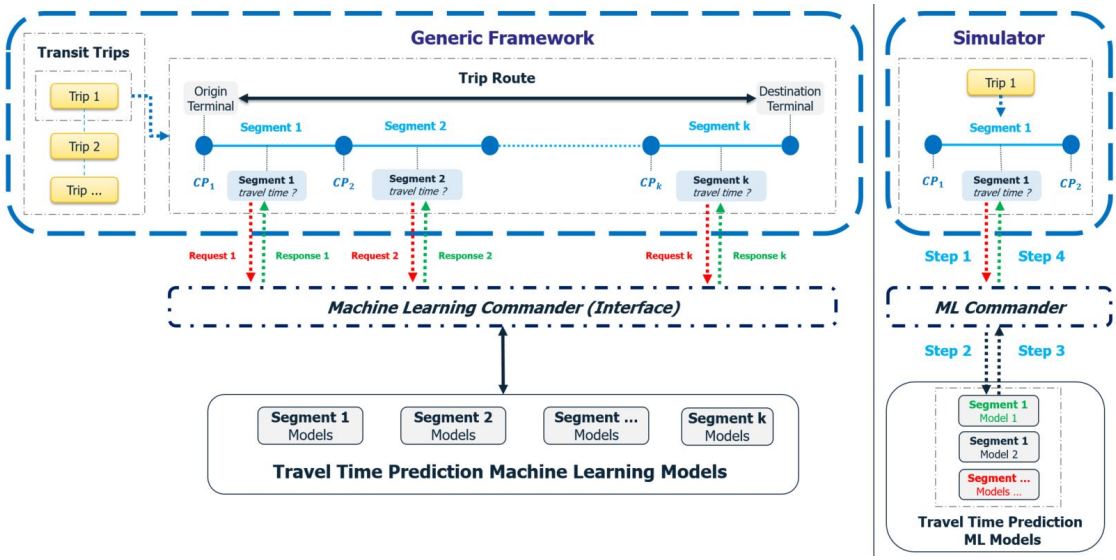

Figure 7: An overview of the proposed coupling process. The simulation model proceeds on trip-by-trip, dividing the trip route into segments according to the defined control points (CPs). At each control point, the travel time of the next segment is estimated through a request-response process. Step1: The simulator requests the segment travel time. Step2: The ML controller selects the appropriate ML model for travel time prediction based on the request information. Step3: The selected model predicts the travel time using the provided information and sends the prediction back to the ML controller. Step4: The ML controller prepares an appropriate response and sends it back to the simulator.

3.2.2. *Coupling Approach: Online Strategy*

In this section, the coupling process will be described in more detail. At this stage of the work, a technical concept that requires further explanation is the simulator request. It consists of three components: 1) Route segment information, including attributes such as transit line, origin, and destination stops of the route segment, which will be used to select the corresponding ML models for travel time prediction; 2) Simulation information, including attributes such as time of day and stop delay, will be mainly used by the selected ML model to predict the travel time of the corresponding segment; 3) The data type variable refers to the type of response data, which may be a single value, such as a mean value, or a probabilistic distribution, such as a normal distribution. An example simulator request is shown in Figure 8.

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <route_segment>
    <feature attribute = "transit_line"     value = "A" />
    <feature attribute = "from_stop"        value = "STP-1883" />
    <feature attribute = "to_stop"          value = "STP-982" />
  </route_segment>
  <simulation_info>
    <feature attribute = "time_of_the_day"  value = "07:41:50" />
    <feature attribute = "stop_delay"       value = "00:00:50" />
  </simulation_info>
  <data_type value = "normal" >
</request>
```

Figure 8: An example of a simulation request, consisting primarily of a route segment and simulation information

The coupling process is performed in several steps as follows:

- The simulator prepares a request and transfers it to the ML interface, including the transit route segment, the simulation information, and the data type of the predicted values.
- The ML commander will use the different request information to find the most suitable ML model. This selection process is carried out in three stages:
  - In the first stage, the ML interface retrieves the deployed ML models from the model database, which match the request segment information. For the sake of simplicity, we assume that there is at least one ML model deployed for each requested route segment.
  - In the second stage, among the retrieved models, the ML models built with mutual features related to the simulation information are selected.
  - In the third stage, the selected deployed models will be compared using evaluation metrics, such as the coefficient of determination R^2. The best-fit model will then be chosen by the ML commander to predict the travel time of the requested segment.
- Next, the chosen model for the travel time prediction will use the simulation information as the test dataset, apply the prediction and then return the corresponding travel time parameters to the ML interface.
- Subsequently, a response will be prepared by the ML commander using the parameters received. Depending on the type of the request, specific processes will be applied to distinguish two cases: 1) the predicted travel time if the requested result is a single value; 2) the parameters of the probabilistic distribution, which can be extracted by fitting the corresponding dataset of the predicted travel time to the requested law.
- Finally, the prepared response will be sent back to the simulator, including the requested parameters.

Figure 9 schematically illustrates the different steps of the coupling process.
In the remainder of the paper, we refer to the ML model as the regression tree model.

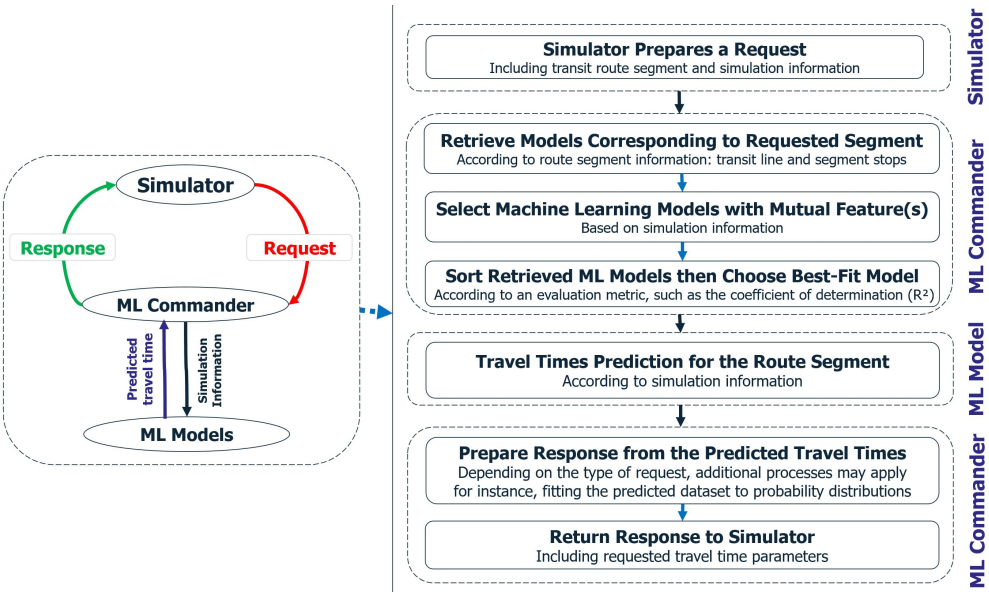

Figure 9: A detailed view of the proposed coupling process. (Left): an illustration of the request-response process, simulator prepares and sends a request to the ML commander, this latter provides the ML models with request simulation information, which will be used to predict the travel time, and this will be returned in response to the simulator afterwards. (Right): coupling process - steps

Travel time prediction by the ML model and response preparation by the ML interface, the most important steps in the coupling process, are presented as follows. First, the simulation information, consisting of a set of feature values, will be used to retrieve the appropriate decision rule, and hence the associated travel time. From a technical point of view, an ML test dataset will be built by matching the ML model features with the simulation values, and will then be used to predict the corresponding decision rule, and therefore the segment travel time. Second, the dataset corresponding to the selected decision rule will be retrieved from the original data. As stated above, depending on the type of the request, the response may take the form of a single value or a distribution. In the first case, the average travel time of the observations will be retained. In the second case, the observations will be fitted with the requested probability law, and the latter parameters will be estimated. Finally, the estimated travel time parameters will be returned to the simulator. An overview of the coupling process with a focus on travel time prediction and response preparation, with the regression tree as the ML model, is shown in Figure 10. In the presented coupling approach, the simulator and ML models exchange simultaneously via the ML interface. Therefore, in the remainder of this paper, the proposed approach will be referred to as the *online strategy*.

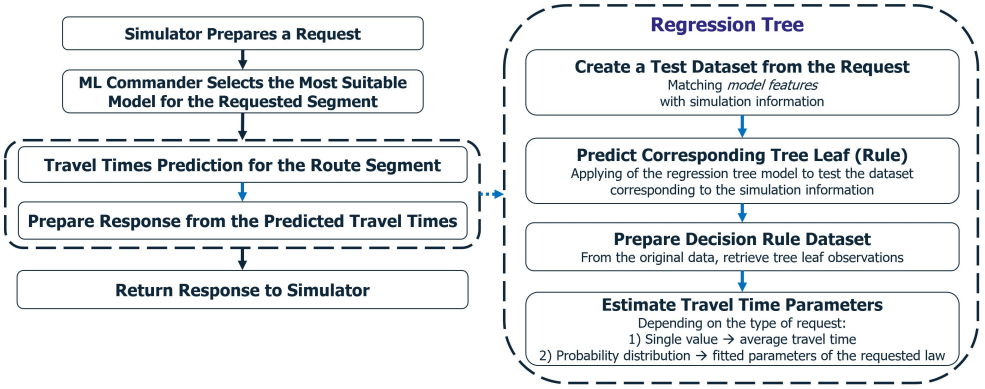

Figure 10: Coupling process - regression tree - the focus is on the stages of predicting travel times and preparing the response

3.2.3. *Coupling Approach: Offline Strategy*

Running a simulation scenario with an online strategy may seem time-consuming because, for each simulation request, a ML model must be selected and then used to predict the segment travel time. To address the drawbacks of the online strategy, an offline coupling strategy is proposed in this section. It aims to run simulation scenarios using approximated segment travel times rather than overusing deployed ML models. In the offline coupling approach, a pre-simulation is first performed, using ML models to create a table of segment travel times, and then integrating the generated table into the framework coupling as part of the ML commander. An overview of the offline coupling strategy is presented alongside the online strategy in Figure 11.

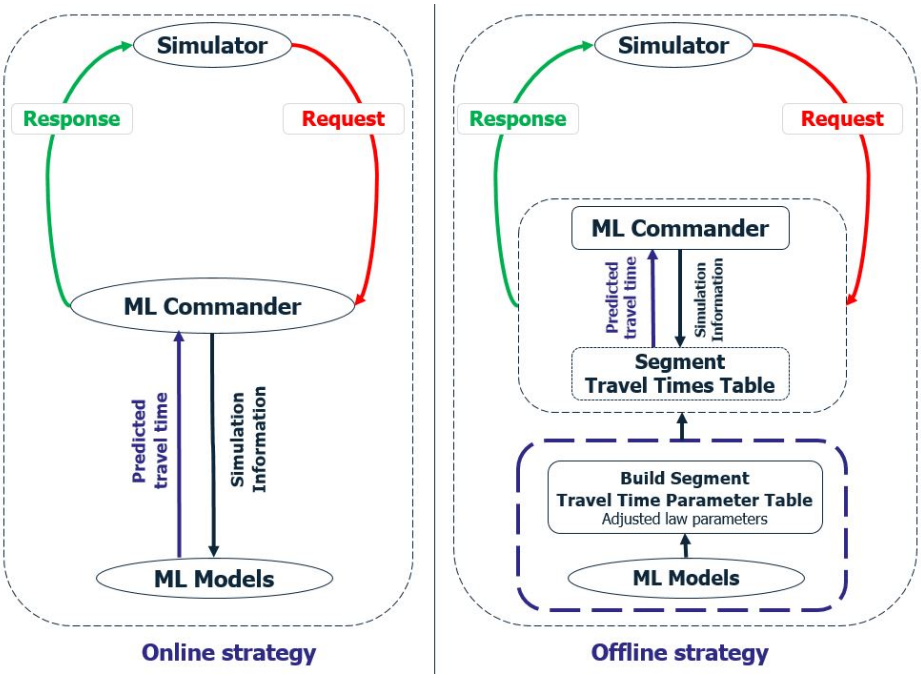

Figure 11: An overview of coupling strategies with emphasis on the request-response process. For the offline strategy, a table with approximate travel time parameters will first be built based on deployed ML models, integrated into the ML interface, and then used to simulate travel times during the request-response process

To build the segment travel time parameter table, the following process will be applied. First, based on a set of features and an evaluation metric, the best-fit deployed ML model will be retrieved for each route segment. Second, a dataset consisting of the actual observations corresponding to each decision rule of the selected model will be prepared. Third, the travel time parameters of the decision rules will be estimated. This involves adjusting the corresponding observed travel times of each decision rule to the chosen probability law. An overview of generating a table of segment travel time parameters based on deployed ML models is depicted in Figure 12.

The presented offline strategy travel times table is built according to the set of decision rules (a set of feature values). A second alternative for estimating a segment's travel time parameter table can be derived based on static time periods (e.g., in 15-minute increments) corresponding to a temporal attribute (e.g., stop time). Therefore, the process of generating the segment travel time parameter table for the second variant is given as follows. First of all, and similarly to the first offline variant, the ML model best suited to the corresponding segment $s \in S$ will be selected. Next, the selected model rules will be filtered, and only a set of rules R corresponding to the time period $t \in T$ will be kept. The fitted parameters of the chosen law will then be estimated for each decision rule $r \in R$. Next, for each selected decision rule $r \in R$, a weight value w_r for period t will be estimated according to Equation 1. Finally, the law parameters per time period $t \in T$ will be estimated according to Equation 2.

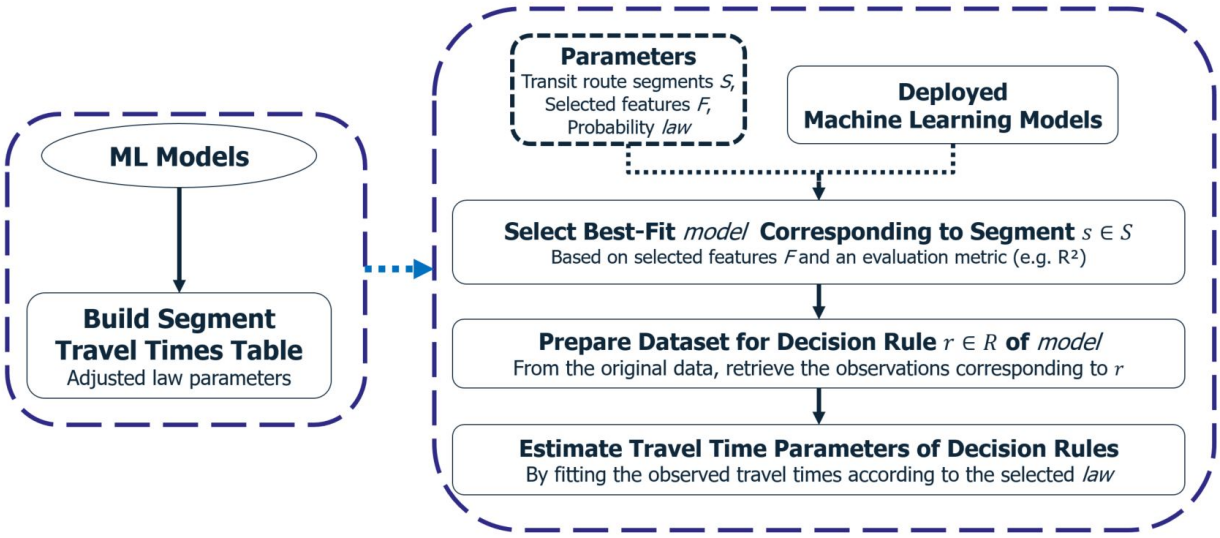

Figure 12: An overview of the building segment travel time parameter table process

$$w_r^t = \frac{|t \cap r_{time\_att}|}{|t|} \tag{1}$$

Where:

- $|t|$: is the absolute duration of a time period t
- r_{time_att}: refers to the selected *time attribute* numeric range in which a decision rule r is satisfied
- w_r^t: is the estimated weight of rule r for time period t

$$P_i^t = \sum_{r \in R} P_i^r \cdot w_r^t \tag{2}$$

Where:

- P_i^t: refers to the estimated value of the law parameter i of the time period t
- P_i^r: refers to the estimated value of the law parameter i of the decision rule r

The normal distribution has been widely used to estimate TTV [15; 14; 76]. An application of Equation 2 according to the normal law is formulated in Equation 3.

$$\begin{aligned} \mu^t &= \sum_{r \in R} \mu^r \cdot w_r^t \\ \sigma^t &= \sum_{r \in R} \sigma^t \cdot w_r^t \end{aligned} \tag{3}$$

Where:

- μ and σ refer to the mean and standard deviation parameters of the normal distribution, respectively.

An overview of the updated process for estimating travel time parameters with static time periods is presented in Figure 13.

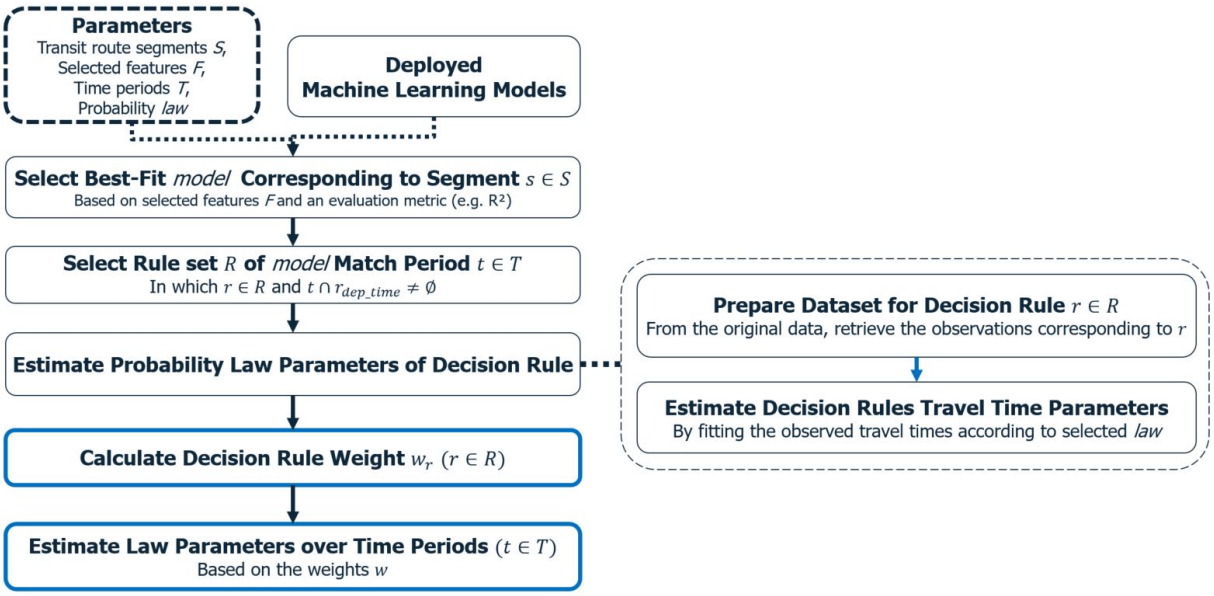

Figure 13: An overview of the process of estimating travel time parameters based on static time periods. The process iterates through each of the route segments and time periods, selects the best-fit segment ML model, retrieves and estimates the corresponding decision rules and their travel time parameters, respectively. Finally, the rule weights will be calculated and then used to deduce the corresponding law parameters over static time periods

After detailing the process of building the segment travel times table, the next step in the offline strategy is to update the request-response process, as follows. First, the simulator prepares its request in the same way as in the online strategy. Then, the ML commander uses the simulation information along with the route segment to retrieve the corresponding travel time parameters from the created table. Finally, the ML commander prepares its response and sends it back to the simulator in the same way as in the online strategy. An overview of the offline request-response process is shown in Figure 14.

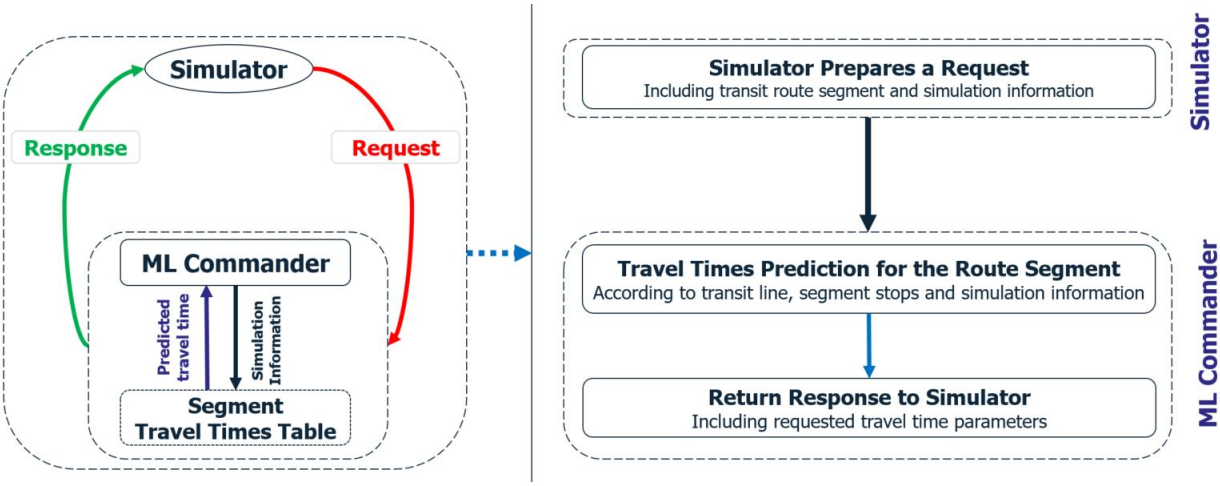

Figure 14: An overview of the request-response for offline coupling strategy. (Left): Overview of the process. (Right): Process steps

3.3. Coupling ML-Simulation: Illustrations

3.3.1. Online Strategy: Illustration

In this section, the application of the described coupling process applied on the first route segment of an anonymous French urban bus line referenced as *A*, will be detailed in the following. The selected transit route and its bus stops are shown in Figure 5. An overview of the coupling framework applied on the chosen route segment is shown in Figure 15.

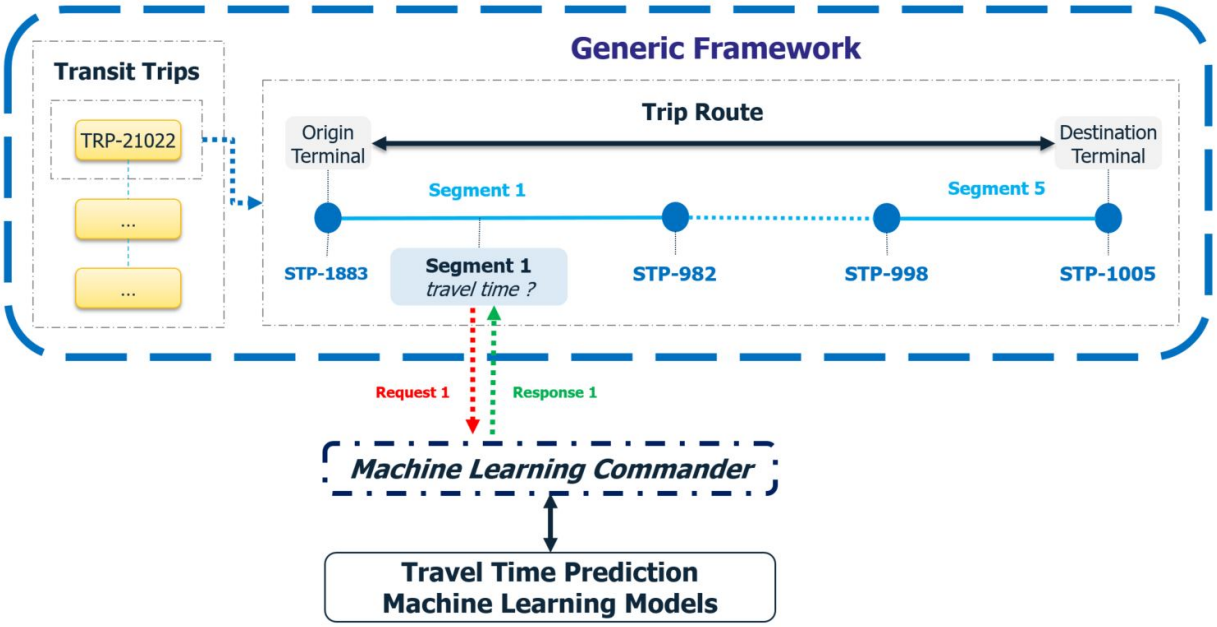

Figure 15: An overview of the coupling framework on a real case application: an anonymous French urban bus line

The first step in the coupling process is preparing the request, which includes: the type of the request, route segment information, and simulation information, the latter including the stop time of the simulated bus and its delay at the stop. An illustration of a simulator request is shown in Figure 8. At this point, it should be noted that the request type, which defines the type of the travel time data returned, will be discussed in more detail below.

Next, the ML commander receives the request from the simulator and processes it in three stages. In the first stage, the segment route information from the request will be used to filter the indexed models. Therefore, the models corresponding to the transit line *A*, with *STP-1883* as the origin and *STP-982* as the destination, will be retrieved. These models are presented in Table 2.

key	algorithm	transit line	from stop	to stop	input features	model target	score (R^2)
68dsfnr6	**RT**	**A**	**STP-1883**	**STP-982**	**stop time**	**real travel time**	**36%**
frvcc9fb	**RT**	**A**	**STP-1883**	**STP-982**	**stop time**	**real travel time**	**35%**
...	...	...	...	...	...	...	...
cq5x4ddu	**RT**	**A**	**STP-1883**	**STP-982**	**stop time;stop delay**	**real travel time**	**36%**
73p81dxl	**RT**	**A**	**STP-1883**	**STP-982**	**stop time;stop delay**	**real travel time**	**35%**

Table 2: A sample of the indexed models with a focus on the selectable route segment models (in bold)

In the second stage, the *time of the day* and *stop delay* attributes will be used to match the suitable ML models. Therefore, ML models without mutual attributes will be dismissed. It is worth noting that request attributes with the same meaning to model input features will be updated. For instance, *stop time* and *time of the day* are two feature attributes with the same meaning. The matched and dismissed models are given in Table 3.

key	algorithm	transit line	from stop	to stop	input features	model target	score (R^2)
68dsfnr6	RT	A	STP-1883	STP-982	stop time	real travel time	36%
frvcc9fb	RT	A	STP-1883	STP-982	stop time	real travel time	35%
cq5x4ddu	RT	A	STP-1883	STP-982	stop time;stop delay	real travel time	36%
73p81dxl	RT	A	STP-1883	STP-982	stop time;stop delay	real travel time	35%

Table 3: A set of ML models from the indexed database matching the simulator request. (Black): ML models corresponding to simulation attributes. (Red): unselected ML models

In the third stage, the selection of the best-fit ML model will be based on evaluation metrics. In this study, the coefficient of determination R^2 is used as an evaluation metric and referenced by *score* attribute in the model index database. Therefore, the selected ML model for travel time prediction, corresponding to the model with the the highest score among the matched models, is given in Table 4.

key	algorithm	transit line	from stop	to stop	input features	model target	score (R^2)
cq5x4ddu	RT	A	STP-1883	STP-982	stop time;stop delay	real travel time	36%

Table 4: The indexed model chosen for travel time prediction

Next, the selected regression tree model will be used to predict the travel time of the illustrated route segment. A test dataset is built by matching the simulation information to the model input features, then used to select the appropriate decision rule, ending with a tree leaf and hence the predicted travel time class. The test dataset, the regression tree used, the rule selected, and its tree leaf are shown in Figure 16.

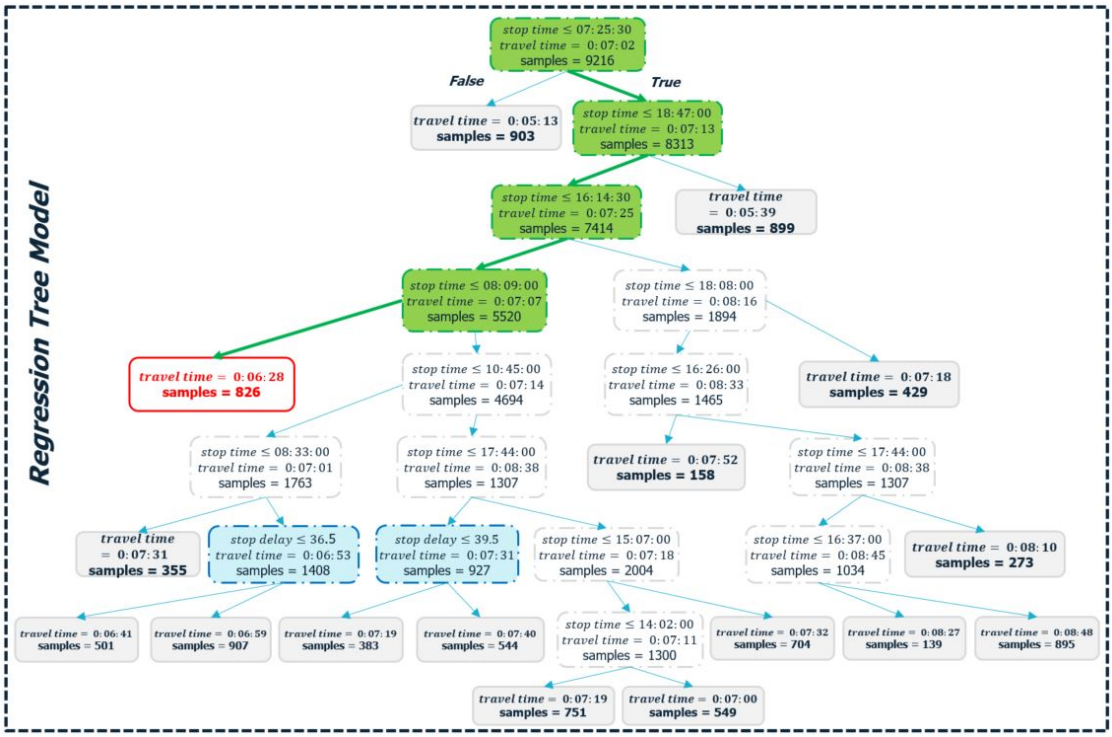

Figure 16: An illustration of applying the chosen regression tree model on the test dataset built on the simulation information. The green nodes constitute the selected path nodes corresponding to the predicted decision rule, the grey nodes are the terminal leaves of the tree, while the red node is the selected tree leaf containing the predicted travel time variable value. On the other hand, white nodes and blue nodes refer to stop time and delay conditions respectively

The last step in the coupling process is the preparation of a response from the predicted travel time. In order to prepare this last response, the selected decision rule observations will be retrieved from the segment's original dataset. Depending on the type of request, two responses are to be considered: 1) Single value request, in which the predicted value is the average travel time of the retrieved observations; 2) Probability distribution request, in which the retrieved observations are fitted to the requested probability distribution. The parameters of the fitted law will therefore be returned. The prepared response, depending on the request type, is shown in Figure 17.

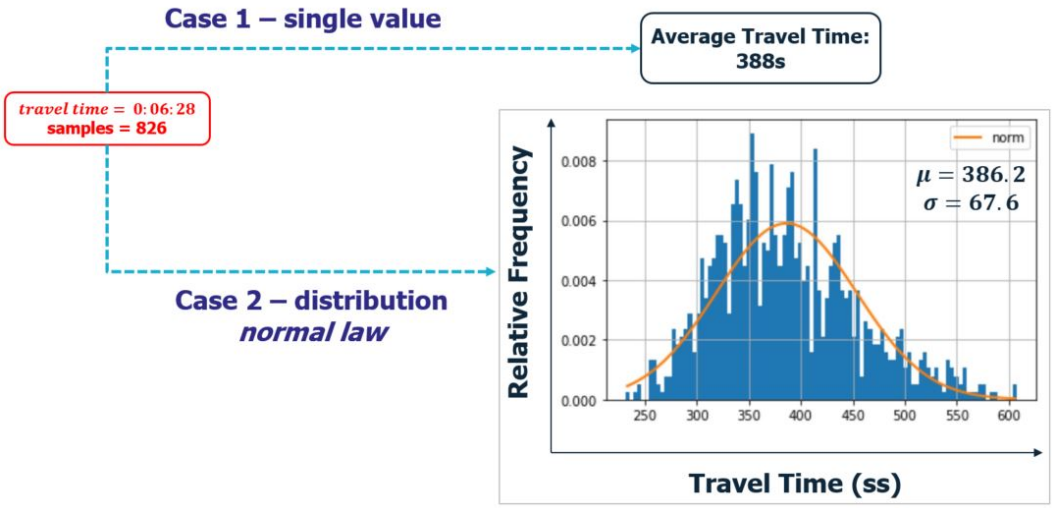

Figure 17: An illustration of the responses prepared based on the type of request. 1) The predicted value is the average travel time of the sample observations. 2) The relative frequency of the retrieved observations is shown, along with the normal law density function requested and its parameters: mean (μ) and standard deviation (σ)

3.3.2. *Offline Strategy: Illustration*

After illustrating an application of the request-response process using the online coupling strategy, this section presents an illustration of the offline strategy. It is carried out in two stages: First, the table of segment travel time parameters, estimated based on the deployed ML models, will be provided; Second, a detailed illustration of a request-response process will be given. The offline strategy will be applied to the same route segment and simulator request shown in the previous section.

To build the segment travel time parameter table, a number of parameters need to be initialized. As stated above, the illustration will take place on the first route segment with *STP-1883* as the origin and *STP-982* as the destination. Depending on the offline strategy variant, the following parameters are initialized as follows. For the first variant, the attributes *stop time* and *start delay* are considered as selected features; the distribution chosen is the *best-fit*, which corresponds to the probability law best suited to the appropriate ML models. For the second variant, the *stop time* attribute is considered as the selected feature; the normal distribution is the chosen law, while the time periods are increments of 15 minutes. The best-fit deployed models selected to create the segment travel time parameter tables are given in Table 5, while the two built segment travel time parameter tables are presented in Table 6 and Table 7, respectively. More details on the travel time parameter tables, as well as the distribution laws, are presented in Appendix A.

key	algorithm	transit line	from stop	to stop	input features	model target	score (R^2)
cq5x4ddu	RT	A	STP-1883	STP-982	stop time;stop delay	real travel time	36%
68dsfnr6	RT	A	STP-1883	STP-982	stop time	real travel time	36%

Table 5: Selected best-fit deployed ML models from the model database. Depending on the offline strategy variant: for the first variant (in black); for the second variant (in blue).

transit line	segment from stop	segment to stop	feature values	probability law	parameter values
...	...	...	...	...	...
A	STP-1883	STP-982	07:25:30 ≤ stop time < 08:09:00	χ^2	loc = 178.97; scale = 11.05; df = 18.83
A	STP-1883	STP-982	08:09:00 ≤ stop time < 08:33:00	exponweib	loc = 113.78; scale = 231.19; a = 4.73; c = 1.96
A	STP-1883	STP-982	08:33:00 ≤ stop time < 10:45:00 ∧ stop delay < 37	genextreme	loc = 388.95; scale = 70.80; c = 0.18
A	STP-1883	STP-982	08:33:00 ≤ stop time < 10:45:00 ∧ stop delay ≥ 37	χ^2	loc = 200.05; scale = 11.13; df = 18.26
...	...	...	...	...	...
A	STP-1883	STP-982	18:47:00 ≤ stop time ≤ 23:59:59	genextreme	loc = 312.36; scale = 65.17; c= 0.16
...	...	...	...	...	...

Table 6: Segment travel time parameter table built for the first offline variant, with *stop time* and *start delay* as selected features and best-fit distribution as the chosen law

transit line	segment from stop	segment to stop	feature values	probability law	parameter values
...	...	...	...	...	...
A	STP-1883	STP-982	07:30:00 ≤ stop time < 07:45:00	normal	$\mu = 384; \sigma = 66$
A	STP-1883	STP-982	08:45:00 ≤ stop time < 08:00:00	normal	$\mu = 384; \sigma = 66$
A	STP-1883	STP-982	08:00:00 ≤ stop time < 08:15:00	normal	$\mu = 404; \sigma = 70$
...	...	...	...	...	...
A	STP-1883	STP-982	18:45:00 ≤ stop time < 19:00:00	normal	$\mu = 432; \sigma = 92$
A	STP-1883	STP-982	19:00:00 ≤ stop time < 19:15:00	normal	$\mu = 336; \sigma = 70$
A	STP-1883	STP-982	19:15:00 ≤ stop time ≤ 19:30:00	normal	$\mu = 336; \sigma = 70$
...	...	...	...	...	...

Table 7: Segment travel time parameter table built for the second offline variant, with *stop time* as the selected feature and normal distribution as the chosen law

After presenting the two segment travel time parameter tables, an illustration of the application of request-response process with the offline strategy will be given. An overview of the framework used with offline as a coupling strategy is shown in Figure 18.

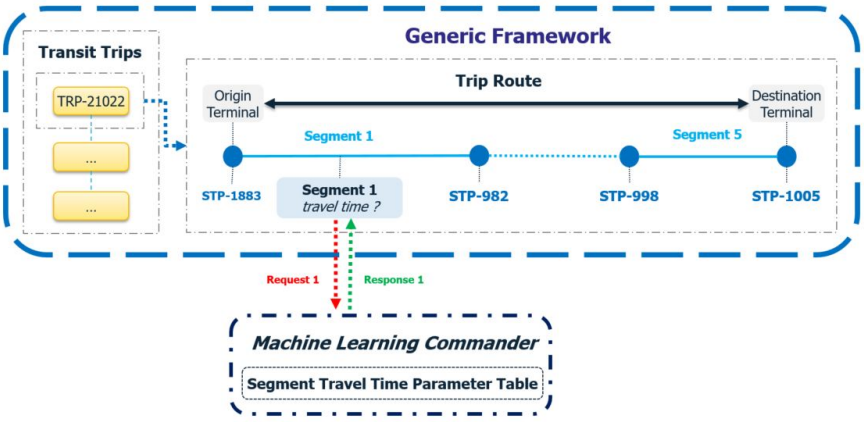

Figure 18: An overview of the framework used with offline as a coupling strategy: a real case of application for an urban bus line

Next, the simulation request presented previously (see Figure 15), will be prepared. Furthermore, in offline coupling, the response is presented in the format of a distribution law with its associated parameters. Therefore, the data type will no longer be included in the request. The updated request is shown in Figure 19.

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <route_segment>
    <feature attribute = "transit_line"      value = "A" />
    <feature attribute = "from_stop"         value = "STP-1883" />
    <feature attribute = "to_stop"           value = "STP-982" />
  </route_segment>
  <simulation_info>
    <feature attribute = "time_of_the_day"   value = "07:41:50" />
    <feature attribute = "stop_delay"        value = "00:00:50" />
  </simulation_info>
</request>
```

Figure 19: Simulation request updated according to offline strategy

Based on the route segment and simulation information, the corresponding decision rule will be selected, by matching the request information with the feature values of decision rule. The corresponding decision rules according to the offline strategy variant are given in Table 8.

transit line	segment from stop	segment to stop	rule feature values	offline variant	parameter values
A	STP-1883	STP-982	07:25:30 ≤ stop time < 08:09:00	first	law = χ^2; loc = 178.97; scale = 11.05; df = 18.83
A	STP-1883	STP-982	07:30:00 ≤ stop time < 07:45:00	second	law = normal; μ = 384; σ = 66

Table 8: Decision rules selected based on offline variant

According to the matched decision rule, a response is prepared by the ML commander, including the corresponding law along with its parameters. An illustration of the responses returned by the ML commander to the simulator, depending on the offline variant, is shown side by side in Figure 20.

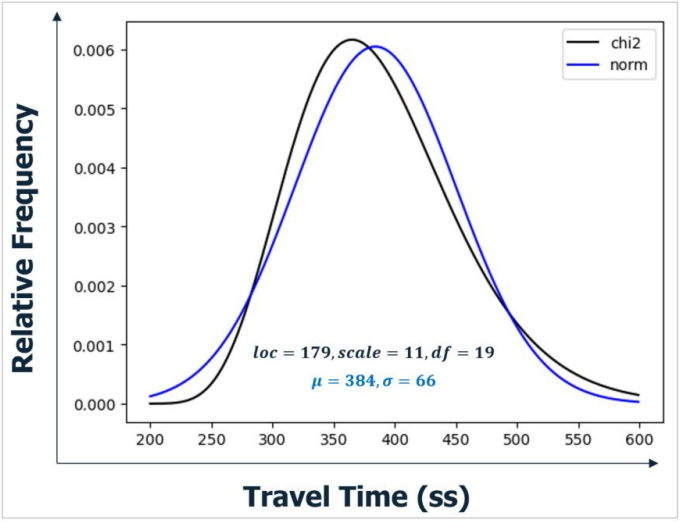

Figure 20: An illustration of the responses returned based on the offline variant, consisting of the selected law density function and its parameters

4. Simulation Scenarios

The simulation scenario was developed for three bus lines referenced respectively as A, B, and C. In this section, the chosen bus lines will be described first, the timing points will be illustrated, and the trained ML models as well as the simulation model parameters will be provided.

4.1. Bus Lines: Description and Timing Points

Line A buses run Monday to Saturday from approximately five in the morning for the first departure and until approximately a quarter past nine in the evening for the last departure. On Sundays, the line's buses run from around six in the morning to around a quarter past nine in the evening. Buses on this line serve 33 stops in one direction and 34 stops in the other direction, over a distance of approximately nine kilometers and a travel time of thirty-five minutes. This bus line is very busy, with an average of one bus every ten minutes during peak hours and one bus every quarter of an hour during off-peak hours, resulting in a total of 179 trips per working day.

Line B operates every day between five in the morning and nine twenty in the evening. Buses on this line serve 32 stops in both directions, covering a distance of approximately nine and a half kilometers and a travel time of just over half an hour. In terms of frequency, Line B runs on average with one bus every six minutes during peak hours and twelve minutes during off-peak hours, resulting in a total of 208 trips per working day.

Line C buses operate all week from approximately five in the morning to nine in the evening, serving 6 stops in one direction and 5 in the opposite direction. Compared to the previous two lines, bus line C is much shorter, with a distance of around two kilometers and a travel time of seven minutes. This line runs on average with one bus every seven minutes during peak hours and ten minutes during off-peak hours, resulting in a total of 186 trips per day.

route ID	**route name**	**route stops**	**route distance (kms)**	**route travel time (minutes)**	**service frequency on/off peak hours**
A_A	*A-Aller*	33	9.00	37	10/15 minutes
A_R	*A-Retour*	34	9.39	38	10/14 minutes
B_A	*B-Aller*	32	9.59	33	6/12 minutes
B_R	*B-Retour*	32	9.67	31	7/12 minutes
C_A	*C-Aller*	6	2.19	6	7/10 minutes
C_R	*C-Retour*	5	1.81	7	7/10 minutes

Table 9: A description of the bus lines studied including: line identifier and name, number of stops per route direction, distance and time travel, in addition to the line's service frequency during peak and off-peak hours. A_A and A_R respectively designate the route in one direction *Aller*, and the route in the opposite direction *Retour* of line A

The public transport simulation scenarios will be carried out on working days (Monday to Friday). The operating days available for the selected lines covered 86, 274, and 281 days for lines A, B, and C, respectively. A description of the bus lines studied is given in Table 9.

After selecting the bus lines for the simulation, the next step is to define the control points for each bus route. These selected points will be used to divide each bus route into route segments, in order to train and then deploy ML segment travel time models. These models will feed the simulation model with appropriate travel times. The retrieved bus timing points, derived route segments, and their scheduled travel times for the selected bus lines are shown in Figure 21.

4.2. Machine Learning: Models Trained

In this work, ML models are trained and then deployed in order to: 1) directly feed the simulation model with the travel times of the route segments in the case of online coupling; 2) build a table of segment travel time parameters in the case of offline coupling. This study focuses on accurately estimating travel time variability rather than returning a single travel time prediction value. ML algorithms such as regression trees, which divide the original dataset into samples based on IF-THEN rules, are suitable for dealing with travel time variability. Thus, the *Regression Tree* algorithm is chosen as the ML algorithm in this study. Next, regression trees are trained with two sets of input features: a) with only *stop time*; b) with *stop time* and *stop delay*. The actual travel time attribute is set as the target.

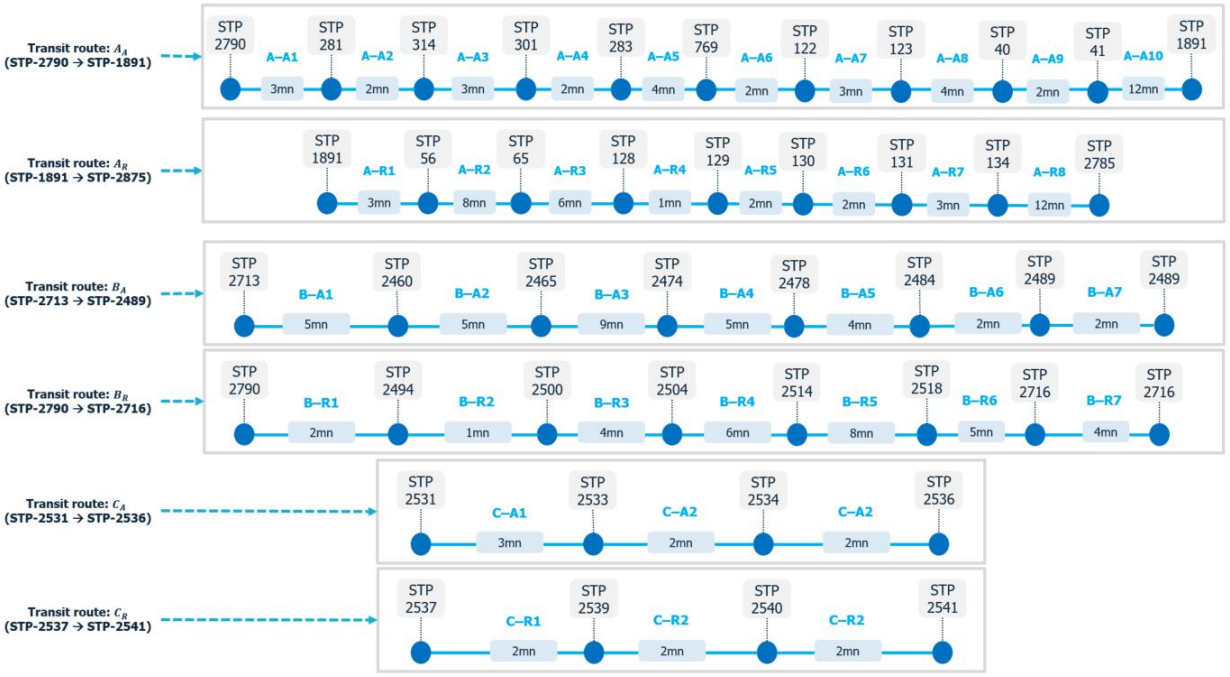

Figure 21: A detailed illustration of the route segments of the three selected bus lines. For bus line A, eleven control points are defined in the *Aller* direction, and nine in the *Retour* direction; there are eight and four timing points per direction for bus lines B and C. The route segment is identified by: its transit line, its route direction (A/R) and its index in the transit route. For instance, *B-R2* refers to the *second* segment of the *Retour* route of line B

To build the ML model, we opt for k-fold cross-validation (CV), a technique used in ML to evaluate the performance of a model on unseen data. In k-fold CV, the dataset is divided into k subsets (known as folds). A fold is used once in each iteration as testing data, while the remaining folds are used as training data [77]. Thus, the process is repetitive until the entire dataset is evaluated. In this study, $k = 5$ is selected.

In terms of parameters, regression trees are built according to the following parameter values: *tree maximum depth* varies between five and fourteen levels, while *minimum samples per leaf* takes four values ranging from 25 to 100. In total, for each route segment, 80 regression trees are built and deployed. For line A, 800 and 640 models were generated respectively, for A_A and A_R. For lines B and C, respectively, 560 and 240 models were generated per route direction.

4.3. Simulation Model: Model and Parameters

In order to evaluate and validate the quality of the different coupling strategies, a simulation model as well as a set of its parameters must be defined. The simulation scenarios will be carried out using an online strategy and three offline coupling variants. In the first two offline variants S_1 and S_2, the segment route travel time parameter table will be built based on the *normal* distribution, with a static time period of a quarter of an hour for S_1. On the other hand, the table of parameters of the third variant S_{3a} will be generated according to the *best-fit* distribution law. We assume that S_{3a} and S_{3b} are considered equivalent during the simulation runtime because no ML models will be trained during the simulation runtime. Thus, the best-fit ML model from which the segment travel times table will be constructed will remain the same. S_{3a} and S_{3b} will be referred to as S_3 in the remainder of this paper. The different coupling variants implemented are given in Table 10.

In this study, the simulation scenarios will be performed using a generic framework, briefly described in section 3.2. A high number of simulation iterations is necessary in order to generate sufficient samples for the evaluation of the coupling strategies. The number of runs to be simulated is set for this study to 1000 iterations.

strategy abbreviation	coupling strategy	probability law	time periods
S_1	offline	normal	15-minutes
S_2	offline	normal	-
S_{3a}	offline	best-fit	-
S_{3b}	online	best-fit	-

Table 10: Implemented coupling variants for the simulated scenario, including three offline strategy variants (S_1, S_2 and S_{3a}) and one online strategy (S_{3b})

5. Experiments and Results

Before presenting the results obtained, the two-step evaluation process undertaken will be briefly detailed. It aims to assess the quality of simulation results obtained from each of the coupling variants using performance indicators. The process is twofold: First, validate ML models for travel time prediction; Second, evaluate the simulated bus on-time performance accuracy of each coupling strategy. From a technical perspective, simulation results will be compared to actual observations, including: 1) segment travel times and 2) bus delays at control points. Furthermore, a computational evaluation of the simulated coupling strategies will take place. Note that the initial dataset was used entirely as a test dataset to validate the simulation performance.

In the first step of the evaluation process, the simulated segment travel times will be compared side-by-side to the actual segment travel times. We denote by TT_A^r and TT_S^r, respectively the actual and simulated average travel times of the route segment r. The two travel times will be compared according to a derived metric $\Delta_{S,A}^r$, which refers to the relative difference per route segment r between TT_S and TT_A, formulated in Equation 4. Furthermore, the two-sample Kolmogorov-Smirnov (KS) test [78], a non-parametric method used to assess the similarity between two distributions, will be performed between actual and simulated travel time distributions, in order to evaluate the resulting day-to-day travel time variability. Additionally, TTV over the course of the day will be evaluated.

$$\Delta_{S,A}^r = \frac{TT_S^r}{TT_A^r} - 1 \tag{4}$$

With $TT_S^r > 0$ and $TT_A^r > 0$

In the second step of the evaluation process, a quality assessment of simulated bus punctuality, using a punctuality measure (operator service reliability), will be performed. It involves comparing actual and simulated punctuality shares side by side. In this study, the punctuality of the operated and simulated buses will be evaluated based on the departure delay at the bus stop at the timing points of each of the bus lines. A bus can be classified according to its delay at the stop into three categories: a) *ahead*, if the bus is *more than one minute ahead of the scheduled time*; b) *on-time travel*, if the delay is between *not earlier than one minute* and *not later than five minutes* at the timing point; and c) *significant delay*, if the bus is *late by more than five minutes* [79].

After introducing the bus punctuality measure, the presented on-time performance metric will be applied at control points to the resulting actual and simulated on-time performance shares. Thus, a Δ deviation measure defined in Equation 5, aimed at quantifying the part of the sharing poorly located between two sets of shares, will be evaluated.

$$\Delta(S_i, S_j) = \frac{1}{2} \cdot \sum_{c \in C} | S_{i,c} - S_{j,c} | \tag{5}$$

Where:

- i and j are two datasets
- C: refers to bus punctuality values with $C = \{ahead, on\text{-}time\ travel, significant\ delay\}$
- S_i: corresponds to the punctuality shares of the buses from the i dataset
- $S_{i,c}$: refers to share of c in S_i with $0 \leq S_{i,c} \leq 1$ and $\sum_{c \in C} S_{i,c} = 1$
- $\Delta(S_i, S_j)$: refers to the part of the sharing poorly located between S_i and S_j with $0 \leq \Delta(S_i, S_j) \leq 1$

5.1. ML: Model Validation

5.1.1. Model Validation: Actual vs Simulated Travel Times

In this section, the first step of the evaluation process will be applied. In order to understand *how well the actual and simulated travel times based on ML models are fitted*, the simulated travel times resulting from the different coupling variants will be compared side by side to the actual travel times. The attributes of the route segments of bus line *A*, including its route segment ID, its segment stops, as well as the actual average (TT_A) and relative metric values Δ, corresponding to the three simulation scenarios (S_1, S_2, and S_3), are given in Table 11.

By analyzing the Δ values, it can be observed that the simulated travel times provided by the ML models match the actual travel times well with a very slight deviation. For most route segments of bus line *A*, Δ does not exceed 5%, with the exception of *A-A2* and *A-R6*. These latter segments are particularly short, at around two minutes of travel time. On the other hand, concerning the simulated travel times of S_1, S_2, and S_3, one can see that the differences are negligible.

Segment ID	Segment Stops	TT_A		$\Delta_{S1,A}$	$\Delta_{S2,A}$	$\Delta_{S3,A}$
A-A1	STP-2790 → STP-281	222		0%	-1%	-2%
A-A2	STP-28 → STP-314	83		-5%	-5%	-5%
A-A3	STP-314 → STP-301	293		-2%	-2%	-2%
A-A4	STP-301 → STP-283	118		-4%	-3%	-4%
A-A5	STP-283 → STP-769	232		-3%	-3%	-3%
A-A6	STP-769 → STP-122	190		-1%	0%	-1%
A-A7	STP-122 → STP-123	86		-4%	-3%	-4%
A-A8	STP-123 → STP-40	248		0%	0%	0%
A-A9	STP-40 → STP-41	91		-3%	-2%	-3%
A-A10	STP-41 → STP-1891	839		-2%	-2%	-1%
-	-	-		-	-	-
A-R1	STP-1891 → STP-56	272		0%	-2%	-1%
A-R2	STP-56 → STP-65	499		-1%	-1%	-1%
A-R3	STP-65 → STP-128	372		0%	-1%	-1%
A-R4	STP-128 → STP-129	67		-2%	-3%	-2%
A-R5	STP-129 → STP-130	130		-2%	-3%	-2%
A-R6	STP-130 → STP-131	120		-5%	-5%	-5%
A-R7	STP-131 → STP-134	243		-3%	-3%	-2%
A-R8	STP-134 → STP-2785	683		2%	2%	2%

Table 11: Reported average travel times TT_A, TT_{S1}, TT_{S2} and TT_{S3} (in seconds) as well as the related derived relative metric values Δ, for each route segment of bus line *A*

After analyzing the different travel times of bus line *A*, the next step is to analyze the second bus line *B* and its travel times. Regarding the Δ values in Table 12, the simulated and actual travel times fit well, with a deviation of no more than 3% for most of segments, except for *B-R1*, *B-R5* and *B-R6*. For *B-R1* and *B-R5*, Δ_{S3} is the highest with 13% and 17% respectively, which is equivalent to a deviation of half a minute and three-quarters of a minute. In total, the simulated travel times from three simulation scenarios are almost identical, with the exception of the departure and arrival segment as well as *B-R5*.

From Table 13, one can see that the route segments of line *C* are relatively shorter than those of the previous two bus lines (*A* and *B*). As for Δ, the relative differences between route segments are minor, averaging a few seconds per route segment. On the other hand, the travel times simulated using S_1 and S_2, correspond statistically well to the actual travel times compared to the simulated travel times of S_3.

Segment ID	Segment Stops	TT_A		$\Delta_{S1,A}$	$\Delta_{S2,A}$	$\Delta_{S3,A}$
B-A1	STP-2713 → STP-2460	288		-2%	-2%	-2%
B-A2	STP-2460 → STP-2465	320		-1%	-1%	-1%
B-A3	STP-2465 → STP-2474	522		-1%	-1%	-1%
B-A4	STP-2474 → STP-2478	411		-2%	-2%	-2%
B-A5	STP-2478 → STP-2484	329		0%	0%	0%
B-A6	STP-2484 → STP-2489	153		3%	3%	3%
-	-	-		-	-	-
B-R1	STP-2490 → STP-2494	219		-7%	-7%	13%
B-R2	STP-2494 → STP-2500	316		-1%	-1%	-1%
B-R3	STP-2500 → STP-2504	357		-3%	-3%	-3%
B-R4	STP-2504 → STP-2514	579		-2%	-2%	-2%
B-R5	STP-2514 → STP-2518	248		0%	0%	17%
B-R6	STP-2518 → STP-2716	320		-6%	-6%	-6%

Table 12: Reported average travel times along with the associated deviation metric values, for each route segment of bus line *B*

Segment ID	Segment Stops	TT_A		$\Delta_{S1,A}$	$\Delta_{S2,A}$	$\Delta_{S3,A}$
C-A1	STP-2531 → STP-2533	184		-2%	-2%	-2%
C-A2	STP-2533 → STP-2534	75		0%	0%	5%
C-A3	STP-2534 → STP-2536	76		-2%	-2%	-2%
-	-	-		-	-	-
C-R1	STP-2537 → STP-2539	102		0%	0%	9%
C-R2	STP-2539 → STP-2540	74		-2%	-2%	-2%
C-R3	STP-2540 → STP-2541	65		-5%	-5%	-5%

Table 13: Reported average travel times as well as Δ values, for the six segments of the *C* bus line

5.1.2. Model Evaluation: Day-to-Day Variability

In order to further analyze the resulting travel times, day-to-day variability will be investigated. Hence, the two-sample Kolmogorov-Smirnov test will be carried out, between on one hand the distribution of actual travel times and on the other hand the distributions of travel times simulated. Cumulative relative frequency distribution curves obtained from the KS test for the three scenarios, over the three lines studied, are given in Figures 22, 23 and 24.

For line *A*, at first glance, negligible differences can be observed between the curves obtained, except for segments *A-A8* as well as the two terminal segments *A-A10* and *A-R8*, in which certain differences are observed. Figure 23 shows slight differences overall, except for *B-A6* and *B-R1*, *B-R5* for S_3, in which a more significant difference between the actual and simulated CDF values is observed. From Figure 24, larger differences are observed, for line *C* compared to those for lines *A* and *B*. These differences can be explained by the particularity of line *C*, on which the route segments are significantly short (around 2km) short routes segments and therefore short travel times.

In order to quantify the observed differences between the actual and simulated CFD, Kolmogorov's D statistic (also called the Kolmogorov–Smirnov statistic) - which quantifies the maximum vertical distance between the empirical distribution functions of two samples - will be analyzed for each route segment. The resulting D-statistic values for each route segment of the three lines studied are shown in Figure 25.

For line *A*, S_3 seems to have the lowest D-statistic among the simulated scenarios on average ($D \approx 0.06$). In contrast, no clear trend is observed for S_1 and S_2, with more similar D-statistic values on average ($D \approx 0.08$). For line *B*, one can observe that S_3 has on average the lowest D-statistic compared to S_1 and S_2. In contrast and surprisingly, S_3 has the highest D-statistic on the segments *B-R1* and *B-R5* (with $D \approx 0.19$). For line *C*, it can be observed that D-statistic values are relatively higher compared to those of lines *A* and *B*. Overall, the results once again show that S_3 has the lowest D-statistic on average compared to S_1 and S_2.

Thus, from the results illustrated on the three bus lines studied, the simulated travel times provided by the ML models are statistically similar to the actual travel times. The travel times simulated with S_1 and S_2 coupling strategies

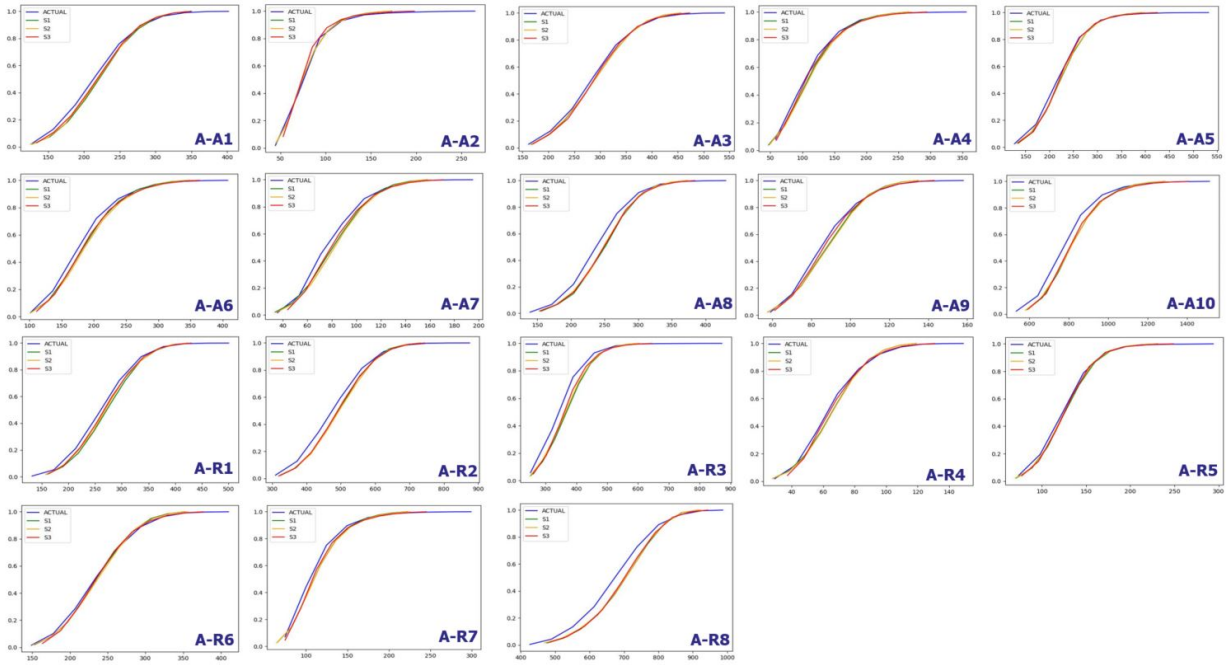

Figure 22: Cumulative relative frequency distribution curves obtained for each route segment of bus line *A*

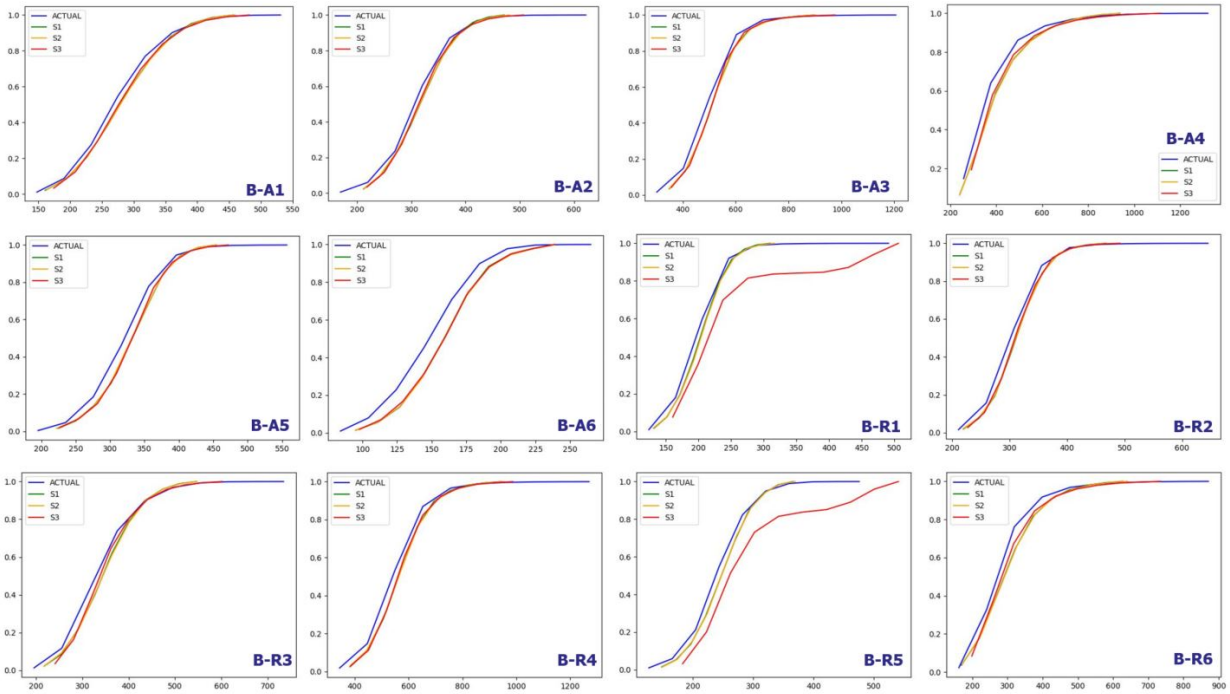

Figure 23: Cumulative relative frequency distribution curves obtained for each route segment of bus line *B*

slightly outperform those generated by S_3.

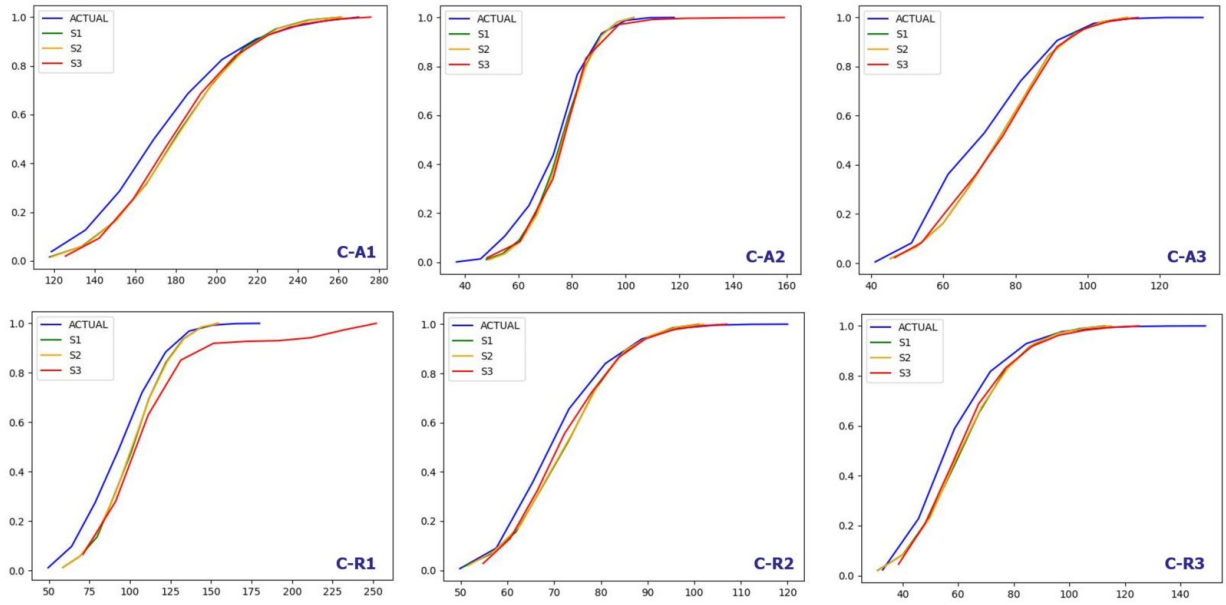

Figure 24: Cumulative relative frequency distribution curves obtained for each route segment of bus line *C*

5.1.3. Model Evaluation: Period-to-Period Variability

After analyzing the day-to-day travel time variability, in this section the focus is on studying the variability of travel times over the course of the day, also known as inter-period or period-to-period variability, which describes the variability between vehicles making similar trips at different times during the same day [74].

From Figure 26, the segments of *A-Aller* route can be grouped into three sets given as follows. The first set consists of segments *A-A1*, *A-A3*, *A-A7*, *A-A8* and *A-A9*, its variation pattern is characterized by two travel time peaks: 1) at noon and 2) at hours during evening rush hour. Furthermore, for most of route segments, a similar trend is observed between actual and simulated travel times. For *A-A1*, the simulation scenarios seem to slightly overestimate the segment travel times. For *A-A3*, a similar tendency is observed during the morning and evening periods, and throughout the day for S_3, while in contrast, S_1 and S_2 appear to follow a different trend and overestimate the travel times of the segments in the afternoon. For *A-A7*, a few observations need to be pointed out. First, the segment travel times are the shortest among *A-Aller* segments. Second, three travel time peaks are observed (morning, noon and evening). Third, the differences between simulated and actual travel times appear larger compared to other segments. For *A-A8* and *A-A9*, an overestimation of the simulated travel times over most of the day is observed, with the simulated and actual travel times fitting well, during the evening peak hours.

The second set consists of segments *A-A2*, *A-A4*, *A-A5* and *A-A6*, its variation pattern is characterized by a constant increase in the travel times of the segments during the day, from early morning until during evening rush hours. Furthermore, for most route segments, a similar trend is observed between actual and simulated travel times. For *A-A2*, the results show that the simulated travel times are slightly underestimated during evening peak-hours. For *A-A4* and *A-A5*, one can observe a very similar tendency between the actual and simulated travel times. For *A-A6*, one can observe that S_1 and S_2 follow different trends, particularly during the inter-peaks period.

The third set, consisting of segment *A-A10*, is characterized by three travel time peaks, during morning and evening rush hours as well at midday. For *A-A10*, the longest route segment, S_3 on the one hand, seems to follow the same trend as the actual observations (three peaks) with differences that seem to be greater in the morning and evening than at midday. S_1 and S_2, on the other hand, experience two peaks (morning and evening), with travel times appearing to stabilize during the inter-peak period.

From Figure 27, for *A-Retour*, the segments can be grouped into two sets as follows. The first set consists of *A-Retour* segments (except *A-A5* and *A-A6*), where travel times during the day are characterized by relatively stable values between the morning and evening rush hours.

Figure 25: Kolmogorov-Smirnov (KS) test results (D-statistic) for each route segment

For *A-R1*, a particular segment with multiple peaks, S_3 on the one hand seems to show similar trends to some extent as the actual observations. S_1 and S_2, however, follow a different trend with two peaks during morning and evening rush hours. For *A-R2*, the simulated travel times follow a similar trend to the actual observations, with S_3 showing a slightly different pattern from S_1 and S_2 in the morning. For *A-R3*, a similar trend is observed with the actual observations for S_3 and, to a lesser extent, for S_1 and S_2. On the other hand, the differences appear most significant among the "A-Retour" segments. Similar trends are observed with negligible differences for *A-R4* and *A-R7*, while more significant deviations were observed during the morning compared to inter-peak times for *A-R8*.

The second set includes *A-R5* and *A-R6*, where a similar trend with a good fit is observed between actual and simulated travel times. Additionally, travel times observed during evening rush hours were significantly higher compared to the rest of the day.

From Figure 28, the segments of the *B-Aller* route can be grouped into two sets as follows. The first set consists of the first three segments, where the actual travel time variation pattern is characterized by two travel time peaks: 1) at midday and 2) during the evening rush hour. The differences between the two peaks are relatively small for segments *B-A1* and *B-A2* and larger for segment *B-A3*. Furthermore, a similar trend is observed for the simulated travel times, which are slightly longer during most of the day, except during the evening peak hours.

The second set, consisting of the last three segments, is characterized by a steady increase in actual travel times throughout the day, with significantly high travel times during the evening rush hours, almost twice those observed in the morning for *B-A4*. For *B-A4*, *B-A5*, and to a lesser extent *B-A6*, the resulting simulated travel times follow a similar pattern to the actual observations with a main peak in the evening, but overestimate travel times throughout most of the day.

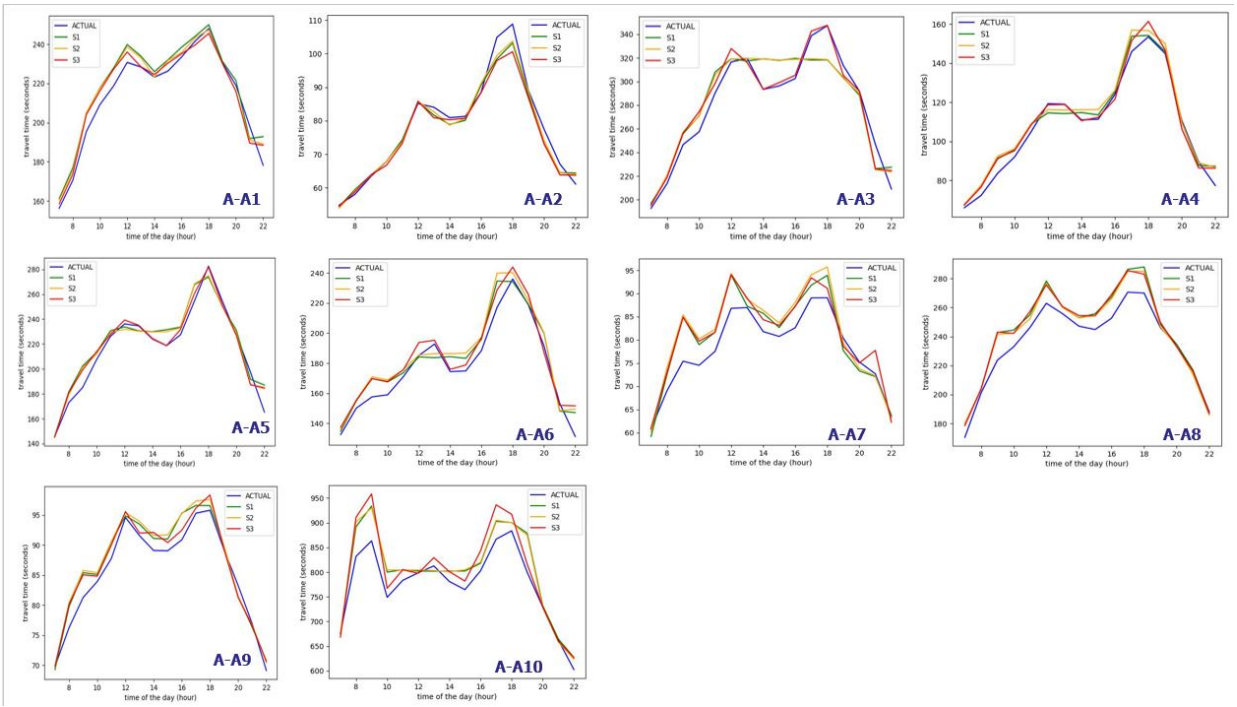

Figure 26: Travel time variability over the course of a day - bus line *A* - Aller

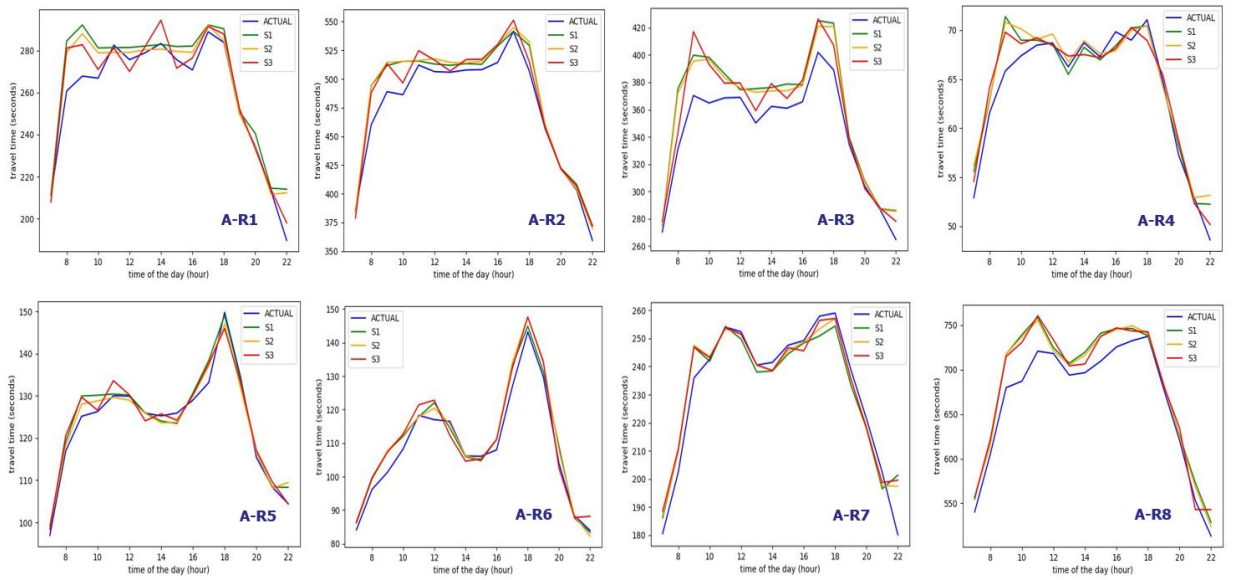

Figure 27: Travel time variability over the course of a day - bus line *A* - Retour

From Figure 29, for *B-Retour*, the segments can be grouped into three sets.

The first set, consisting of *B-R1* and *B-R5*, is characterized by a small variation in actual travel times over the course of the day. Additionally, the travel times of S_1 and S_2 appear to follow the same pattern as the actual travel times with minor deviations. In contrast, the travel times of S_3 seem significantly deviated, with substantial overestimation during the evening and afternoon for *B-R1* and early in the morning for *B-R5*.

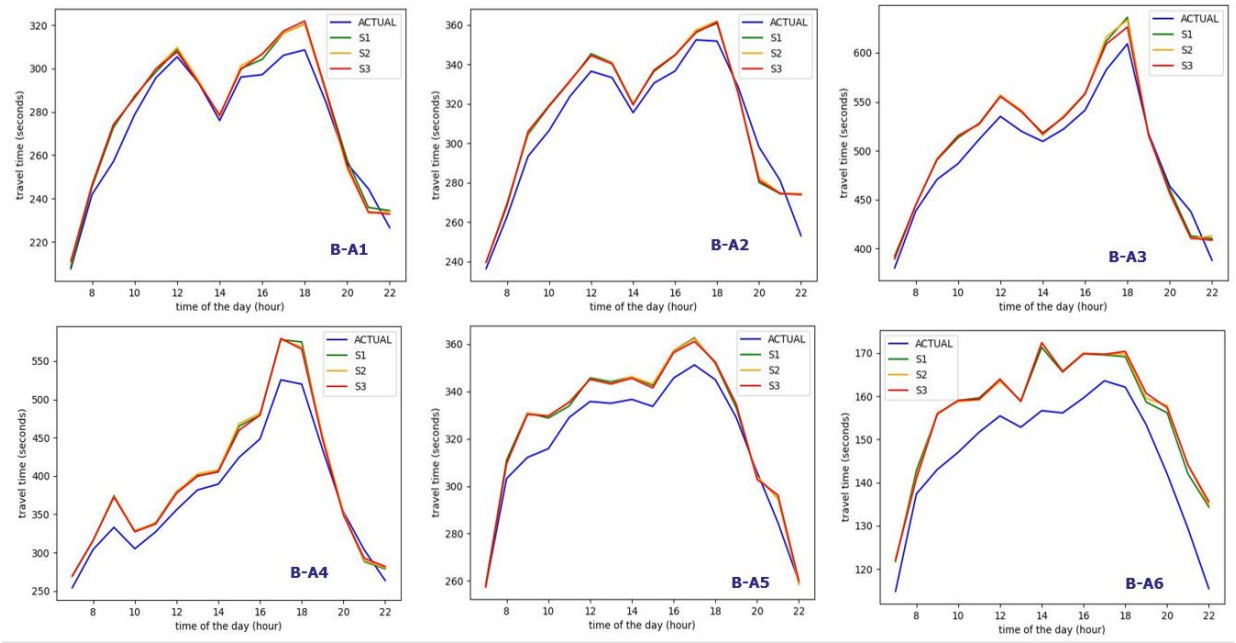

Figure 28: Travel time variability over the course of a day - bus line *B* - Aller

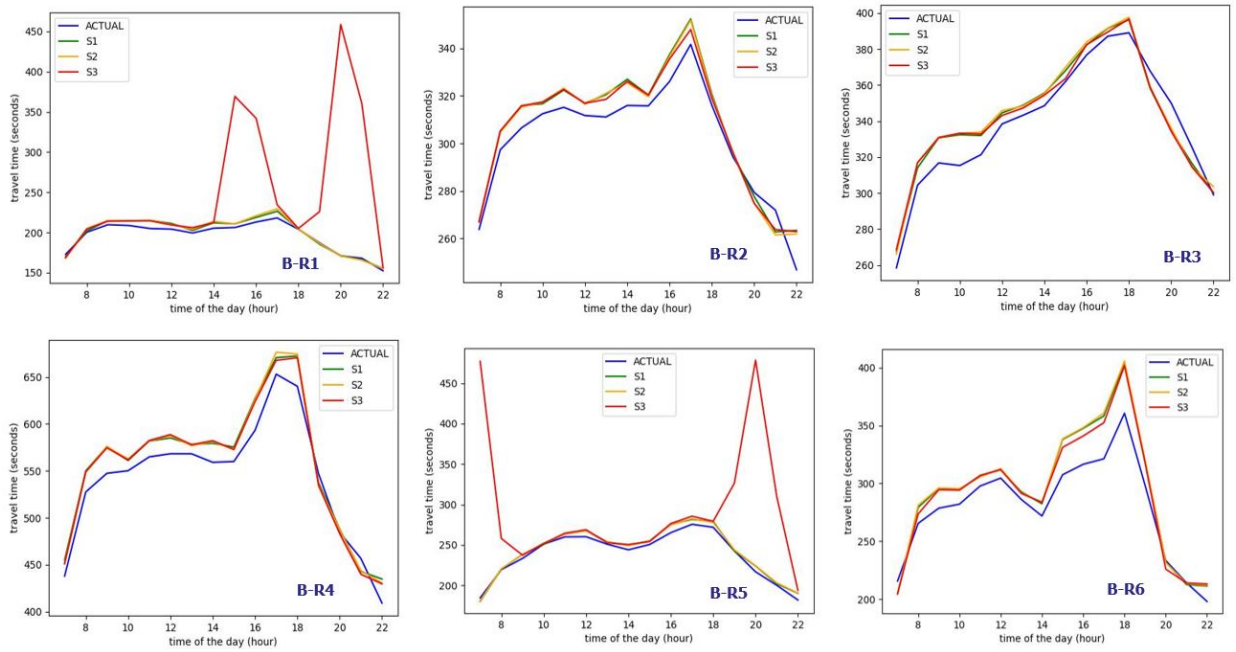

Figure 29: Travel time variability over the course of a day - bus line *B* - Retour

The second set includes segments *B-R2*, *B-R3*, and *B-R4*. This set is characterized by a steady increase in travel times during the day for *B-R3* and *B-R4* and, to a greater extent for *B-R2*, with a peak in actual travel times during the evening rush hours. A similar trend is observed for the simulation scenarios, with travel times overestimated to some extent throughout the day, except during the evening.

The third set consists of segment *B-R6*, where two peaks can be observed at midday and during the evening rush hours, with relatively small differences between the two peaks. Additionally, the simulated travel times seem slightly overestimated.

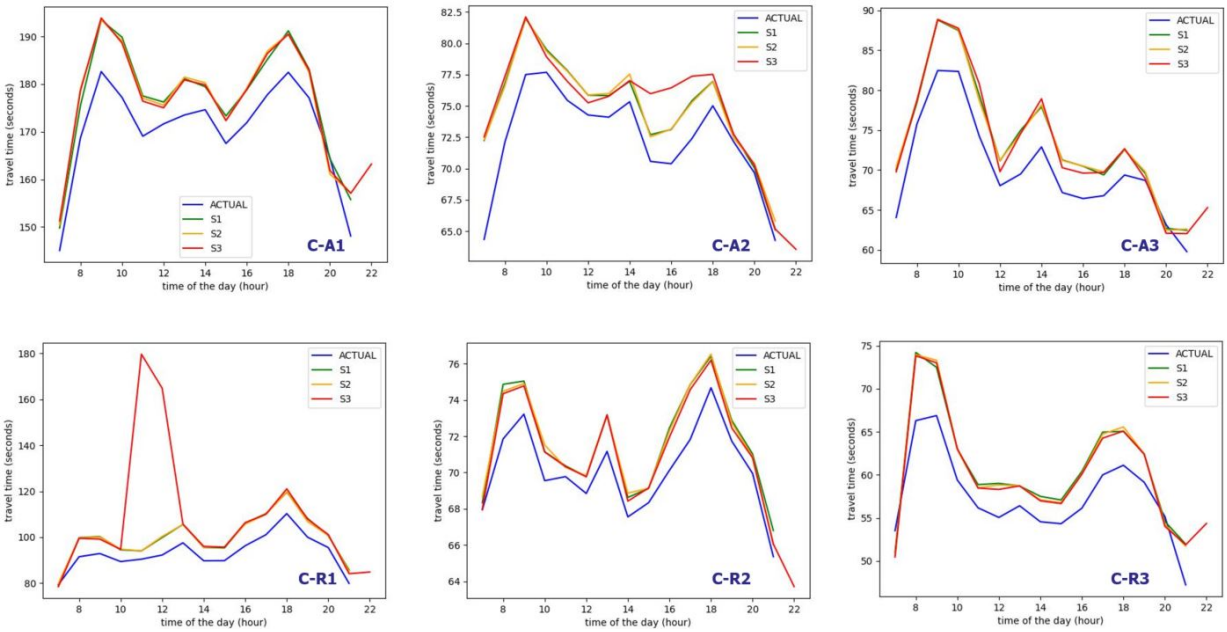

Figure 30: Travel time variability over the course of a day of bus line *C*

After analyzing the period-to-period travel time variability of lines *A* and *B*, the next step is to analyze the variability of line *C*, focusing specifically on the departure segments (*C-A1* and *C-R1*) with an average travel time of more than two minutes.

From Figure 30, similar trends between actual and simulated travel times can be observed, including multiple peaks (morning, noon, evening). Contrary to the trends observed for lines *A* and *B*, travel times during morning rush hours are longer than during evening rush hours. Furthermore, the differences appear relatively minor, averaging just a few seconds. However, a notable observation is the trend for S_3, where the simulated travel times are significantly overestimated compared to the actual travel times for *C-A2* and *C-R1*, particularly in the afternoon and around noon.

5.2. *Simulation Model Precision: Actual vs Simulated Bus Punctuality*

The second step in the evaluation process is to assess simulated bus punctuality. The punctuality measures outlined in Section 5 will be estimated based on actual observations and the three simulated scenarios (S_1, S_2, and S_3) for each of the three bus lines (*A*, *B*, and *C*). The resulting shares, obtained from each dataset, will be statistically analyzed and compared side by side.

Figure 31 provides initial insights into how bus on-time performance is distributed according to the presented metric, applied to all 12 datasets. Further details on the differences between the shares resulting from the simulation scenarios and the actual shares are provided in Table 14. At first glance, one can observe some differences in the shares between the different bus lines, with *A* and *C* appearing to be slightly more punctual than *B*. Based on *ahead* shares, line *C* buses are much *more ahead* compared to lines *A* and *B*. Furthermore, line *C*, with its shortest bus routes, has the least number of late buses, where the share of *significant delay* is negligible, while lines *A* and *B* have statistically similar ratios for significantly delayed buses.

For bus line *A*, results from actual observations show that approximately two-thirds of buses are on time and about one-fifth are early at control points. On the other hand, the shares obtained from the three simulated scenarios indicate a very similar trend, with minor differences (± 2%) across different delay categories. In total, the simulation shares appear to be comparable to the actual shares, with differences ranging from 4% to 5%, resulting from a slight underestimation of the share of buses with significant delays.

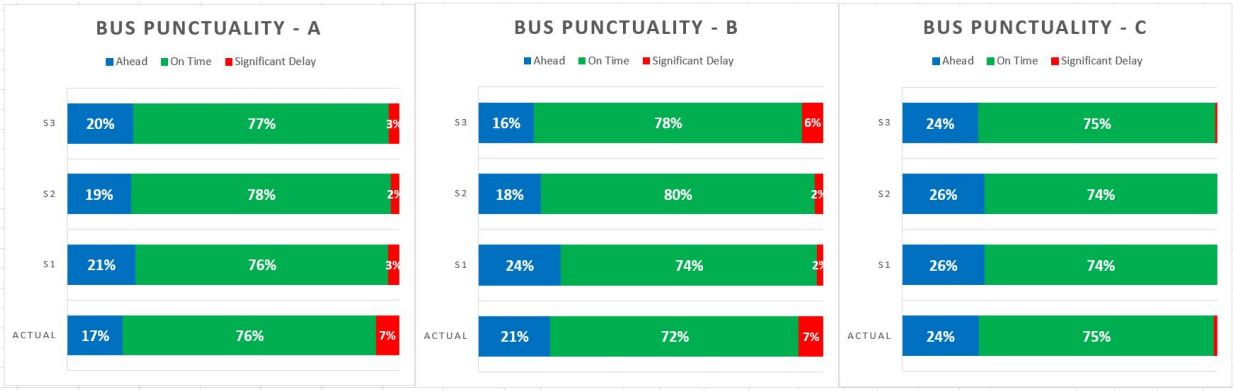

Figure 31: Bus punctuality shares for the bus lines studied, based on the first punctuality measurement. For the three studied lines, a strong similarity shares is observed between S_1, S_2 and S_3 compared to actual shares, with S_1 appearing to slightly outperform S_2 and S_3

For bus line B, approximately two-thirds of actual buses are *on-time travel* and one-fifth are *ahead*. Buses with a significant delay represent less than a tenth of the total buses. Furthermore, the results show that the resulting shares do not follow exactly the same trend. While S_1 slightly underestimates the share of late buses (-5%), S_2 and S_3 tend to overestimate the share of *on-time travel* (+8% and +6%, respectively). As with bus line A, the differences between the actual and simulated shares are moderate, ranging from approximately 5% to 8%.

According to the results obtained from applying the first metric to the bus line C datasets, approximately two-thirds are on time and one-quarter are early. On the other hand, the share of buses with significant delays is very negligible. Unlike the two previous bus lines (A and B), the resulting shares from the three simulated scenarios show an identical trend to the actual shares with negligible differences not exceeding 2%.

In summary, the analyzed results show significant similarities between the shares of S_1, S_2, and S_3 with minor differences on bus lines A and C, and to a lesser extent on line B. Compared to actual bus delays, the delay shares resulting from the S_1 simulation scenario appear to be slightly more accurate than those from S_2 and S_3, which slightly overestimate the share of *on-time* buses. In short, the resulting simulation distributions for the three studied lines are well-fitted to the actual reference distributions according to this metric, demonstrating good accuracy of the simulation model.

	Line A				Line B				Line C			
	S_a	S_1	S_2	S_3	S_a	S_1	S_2	S_3	S_a	S_1	S_2	S_3
Ahead	17%	21%	19%	20%	21%	24%	18%	16%	24%	26%	26%	24%
On-time travel	76%	76%	78%	77%	72%	74%	80%	78%	75%	74%	74%	75%
Significant delay	7%	3%	2%	3%	7%	2%	2%	6%	1%	0%	0%	1%
$\Delta(S, actual)$	-	**4%**	**5%**	**5%**	-	**5%**	**8%**	**6%**	-	**2%**	**2%**	**0%**

Table 14: Illustration of the differences and gaps between the resulting simulation shares and the actual shares according to the first punctuality metric, for lines A, B and C

5.3. *Coupling Model Performance: Offline vs Online Computation Time*

After validating the simulated travel times and evaluating the accuracy of the implemented coupling strategies, the next step is to evaluate the simulation model performance. Figure 32 illustrates the experimental computation times for building a decision tree for travel time prediction as well as simulation execution time.

From Figure 32.(a) and (b), one can observe that the computation time for building one ML model, which includes dataset preparation and training time, is higher for lines B and C compared to line A. On the other hand, it can be seen that the total time required to build all segment models is higher for line A than for lines B and C. These trends can be explained by a larger dataset for bus lines B and C, and a larger number of route segments for line A.

From Figure 32.(c) and (d), the simulation time appears to be slightly higher for lines A and B compared to line C. For the coupling strategies, S_1 is notably the strategy with the lowest simulation time for all three bus lines. On average, S_1 requires two to three times less computation time compared to S_2 and S_{3a}, and up to seven times less than S_{3b}. For S_2 and S_{3a}, which are based on the *normal* and *best-fit* probability laws, respectively, a similar execution time can be observed, with S_{3a} having a relatively higher simulation time than S_2. As expected, for S_{3b}, the online coupling strategy has a high computational time compared to the three offline coupling variants. For the scenarios conducted, the online strategy simulation is on average twice as slow as its equivalent offline strategy.

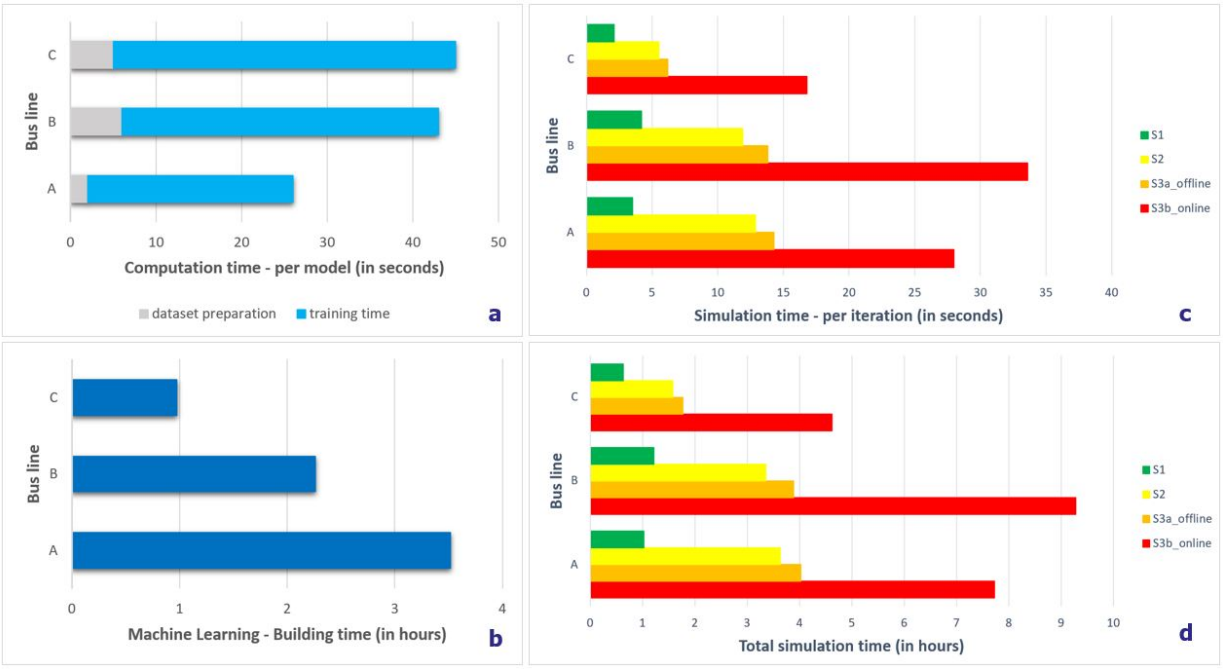

Figure 32: Overview of the different calculation times for each bus line. (a): Average time to prepare the dataset and train tree regression for segment travel time prediction. (b): Total ML model building time per bus line. (c) and (d): Single and total simulation time for each bus line (1000 iterations). S_1 is the strategy with the lowest calculation time, while the online strategy S_{3b} is the one with the highest computation time

The resulting differences in total simulation time between the studied lines can be explained by: 1) the number of route segments and 2) the number of simulated trips. The advantage of S_1 over other strategies is explained by the fixed size of the table of segment travel time parameters, built on the basis of a feature (stop time) and a period static time (15 minutes). Furthermore, the relative difference in calculation time between S_2 and S_{3a} could be explained by the fact that the empirical time necessary to generate a valid travel time value with the normal law is, on average, lower than that of the best-fit law. Additionally, the application of the request-response during the simulation is time-consuming, particularly using the online coupling strategy, which requires retrieving, for each request, the corresponding ML model from the indexed ML database. Therefore, excessive access to the index database leads to a much longer simulation time, as in S_{3b}. It should be noted that applying an offline strategy requires building a table of travel time parameters, with the building time being relatively minor compared to the scenario simulation time, ranging from a few seconds to a few minutes depending on the number of segments and the models built, and to a lesser extent on the distribution law chosen.

6. Discussions

The simulated travel times provided by the ML models match well with the actual travel times on the three lines studied, with S_1 and S_2 slightly outperforming S_3. From Table 12, S_3 appears to significantly overestimate the travel times of *B-R1* and *B-R5*, which can be explained by an imprecise trend in period-to-period travel time variability. This results in a significant deviation in travel times, in the afternoon and evening for *B-R1* and during off-peak hours in the morning and the evening for *B-R5*.

Concerning day-to-day variability, the differences between the actual and simulated CFD for segments *A-A10* and *A-R8* (see Figure 22) are the result of a significant overestimation of travel times during morning rush hours and, to a lesser extent, during evening peak hours, in which travel times are the longest. Similarly for *B-A6* (see Figure 23), and despite a relatively small deviation between the travel times obtained ($\Delta \approx 3\%$), the differences can be explained by a relative overestimation of the simulated travel times, particularly during off-peak hours in the morning and evening.

Concerning period-to-period variability, three trends were observed: 1) A pattern with a peak in the evening, which can be explained by the impact of the increase in traffic over the course of the day; 2) A pattern with two peaks at midday and in the evening, which can be explained by a reduction in road congestion in the afternoon; and 3) A pattern with three peaks (morning, noon, and evening), for certain segments of line *C*, where the earliest peak is explained by the presence of traffic lights as well as crossroads near educational facilities such as high schools. Furthermore, the results showed that travel times during evening rush hours are significantly longer than the rest of the day, which can be explained by a high level of traffic and congestion.

For S_1 and S_2, a very similar trend was observed, which can be explained by the choice of normal distribution to fit the observed travel times. The travel times estimated using S_1 and S_2 were more regular during the inter-periods, which can be explained once again by: 1) the use of normal law and 2) a relatively small variation in the observed travel times. In contrast, the S_3 trends appear to match those in actual travel times on most route segments. However, it also fails to accurately estimate travel times, especially during inter-periods, leading to discrepancies, which can be explained by an unsuitable law for travel time fitting.

The delay shares resulting from the simulation generally corresponded well to the actual shares, with a slight underestimation of the shares of late buses according to the first bus punctuality metric. Furthermore, none of the applied coupling strategies seem to stand out; the differences appear very limited or even negligible, S_1 and S_3 being slightly more regular and outperforming S_2. On the other hand, according to the second punctuality metric, the resulting shares of the simulation lack precision, with significant differences, particularly for buses on time and slightly late. This could be explained by a late departure of some buses. For instance, if a bus leaves its terminal relatively late, it may not be able to make up the scheduled stop time and, therefore, be late at control points, resulting in an increase in the number of delayed buses. To overcome this drawback, integrating actual bus departure delays into the simulation model will greatly improve the simulation results.

From a computational point of view, the bus-line simulation time scales depending on the number of route segments and trips simulated. S_1 is the strategy with the least simulation time, followed by S_2 and S_{3a}, while S_{3b} has the highest simulation time. The advantage of using an online strategy is retrieving from an updated model database. Additionally, it provides the ability to use multiple models per route segment. On the other hand, applying an offline strategy allows the use of a single model per route segment, with the need to rebuild the parameter table each time a new ML model is built and inserted into the database. The building time of the travel time parameter table increases slightly each time a new model is indexed but remains much lower than the online simulation time. In summary, applying an offline coupling strategy requires less simulation time than an online strategy.

Overall, S_1 seems to stand out from other coupling strategies with a good trade-off between simulation time and adequacy to actual observations for both travel time and bus punctuality. In short, the online coupling strategy is more suitable for real-time applications. On the other hand, offline coupling strategies, with good model accuracy and the best computation time, are suitable for simulation applications.

7. Conclusions

This paper sheds light on how ML and public transport simulation models can be effectively coupled. In this paper, we suggested two coupling strategies: 1) online, in which ML models provide PT simulators with travel times, and 2) offline, in which travel times resulting from ML are integrated as input data into the transit simulation model. In the proposed framework, various data sources are integrated, ML models are trained and deployed for predicting bus travel times, and simulation scenarios are implemented and executed. A case study was carried out on three bus lines in a French city with four scenarios, including three offline coupling variants (S_1, S_2, and S_{3a}) and one online variant (S_{3b}) per bus line.

A first analysis aimed at validating the simulated travel times was carried out. Furthermore, two analyses were conducted to further evaluate the variability in travel from day to day and from period to period. The results showed that S_1 and S_2 followed a very similar trend, while S_3 matched the actual trends but also showed difficulties in accurately predicting travel times in particular situations, leading to some deviations. Overall, the resulting simulated travel times adequately match actual travel times, with slight differences observed, in which S_1 and S_2 slightly outperform S_3. On the other hand, an evaluation of the accuracy of the model was carried out. Despite the use of more approximate travel times, the offline strategies show as good a match as the online strategy between the actual and simulated bus punctuality shares. In terms of performance, S_1 stood out from other coupling strategies with less simulation time. Overall, S_1 appears to be the best strategy, with good model precision and the best calculation time. We believe this paper is of particular interest for practitioners in the field, as it provides insights into how to effectively couple ML and public transport simulation models, as well as the benefits and limitations of each strategy.

Author Contributions: Investigation, Y.D.; Methodology, Y.D. and O.C; Supervision, O.C. and M.N.; Writing—original draft, Y.D.; Writing—review and editing, O.C, M.N. and M.H.; Funding acquisition, M.H. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgement: The authors thank the A1 Statistiques company and its operator for making the data available. They also thank Vincent Leblond of the Tellae company and François Queyroi and Gabriel Ferrettini of DUKe, a research team of the Laboratory of Digital Sciences of Nantes (LS2N), for their help for structuring and analysing the data. They also thank the region of Pays de la Loire for SIMULBUS grant.

Appendix A.

Abbreviation	Description
χ^2	Chi-squared distribution
dweibull	Double Weibull distribution
exponnorm	Exponentially modified Gaussian distribution
exponweib	Exponentiated Weibull distribution
gamma	Gamma distribution
genextreme	Generalized extreme value distribution
log-normal	Log-normal distribution
normal	Normal distribution
rayleigh	Rayleigh distribution

Table A.15: List of probability laws used in this work

feature values	probability law	parameter values						
		loc	scale	a	c	df	K	s
00:00:00 ≤ stop time < 07:25:30	exponnorm	279.04	47.24				0.72	
07:25:30 ≤ stop time < 08:09:00	χ^2	178.97	11.05			18.83		
08:09:00 ≤ stop time < 08:51:00	genextreme	405.45	81.22		0.16			
08:51:00 ≤ stop time < 09:52:30	log-normal	63.20	328.82					0.21
09:52:30 ≤ stop time < 10:45:00	genextreme	397.87	61.38		0.17			
10:45:00 ≤ stop time < 11:05:00	rayleigh	286.08	126.84					
11:05:00 ≤ stop time < 12:32:00	genextreme	420.60	69.58		0.11			
12:32:00 ≤ stop time < 15:02:00	log-normal	-43.50	470.37					0.14
15:02:00 ≤ stop time < 16:09:00	genextreme	416.49	70.86		0.13			
16:09:00 ≤ stop time < 16:26:00	genextreme	432.99	87.79		0.15			
16:26:00 ≤ stop time < 17:32:00	gamma	221.73	57.94	5.32				
17:32:00 ≤ stop time < 17:51:00	gamma	183.22	45.78	7.23				
17:51:00 ≤ stop time < 18:14:00	rayleigh	275.95	165.22					
18:14:00 ≤ stop time < 18:58:30	genextreme	349.26	41.30				1.91	
18:58:30 ≤ stop time ≤ 23:59:59	χ^2	66.03	9.13			29.57		

Table A.16: An illustration of the built travel time parameter table for the route segment, in which *from stop = STP-1883* and *to stop = STP-982* of transit line *A*, with *stop time* as selected feature and best-fit distribution as the chosen law. The meaning of the law parameters can be found in **https://docs.scipy.org/doc/scipy/tutorial/stats/**

feature values	probability law	parameter values						
		loc	scale	a	c	df	K	s
00:00:00 ≤ stop time < 07:25:30	exponnorm	279.04	47.24				0.72	
07:25:30 ≤ stop time < 08:09:00	χ^2	178.97	11.05			18.83		
08:09:00 ≤ stop time < 08:33:00	exponweib	113.78	231.19	4.73	1.96			
08:33:00 ≤ stop time < 10:45:00 ∧ stop delay < 37	genextreme	388.95	70.80		0.18			
08:33:00 ≤ stop time < 10:45:00 ∧ stop delay ≥ 37	χ^2	200.05	11.13			18.26		
10:45:00 ≤ stop time < 12:32:00 ∧ stop delay < 40	genextreme	406.98	64.48		0.06			
10:45:00 ≤ stop time < 12:32:00 ∧ stop delay ≥ 40	gamma	137.61	19.89	16.24				
12:32:00 ≤ stop time < 14:02:00	exponnorm	397.84	53.72				0.78	
14:02:00 ≤ stop time < 15:07:00	exponnorm	372.67	50.53				0.96	
15:07:00 ≤ stop time < 16:14:30	gamma	135.73	19.93	15.81				
16:14:30 ≤ stop time < 16:26:00	dweibull	469.40	85.49		1.34			
16:26:00 ≤ stop time < 16:37:30	dweibull	502.70	100.90		1.33			
16:37:30 ≤ stop time < 17:44:00	exponweib	-403.20	120.65	2346.43	1.03			
17:44:00 ≤ stop time < 18:08:00	χ^2	245.01	28.04			8.80		
17:44:00 ≤ stop time < 18:47:00	exponnorm	357.21	42.61				2.00	
18:47:00 ≤ stop time ≤ 23:59:59	genextreme	312.36	65.17		0.16			

Table A.17: An illustration of the built travel time parameter table for the route segment, in which *from stop = STP-1883* and *to stop = STP-982* of transit line *A*, with *stop time* and *start delay* as selected features and best-fit distribution as the chosen law. The meaning of the law parameters can be found in **https://docs.scipy.org/doc/scipy/tutorial/stats/**

References

[1] K. Small, E. T. Verhoef, The economics of urban transportation, Routledge, 2007.

[2] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: concepts, methodologies, and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 5 (3) (2014) 1–55.

[3] T. P. L. Le, T. A. Trinh, Encouraging public transport use to reduce traffic congestion and air pollutant: A case study of ho chi minh city, vietnam, Procedia engineering 142 (2016) 236–243.

[4] L. R. Abdulrazzaq, M. N. Abdulkareem, M. R. M. Yazid, M. N. Borhan, M. S. Mahdi, Traffic congestion: Shift from private car to public transportation, Civil Engineering Journal 6 (8) (2020) 1547–1554.

[5] T. Reich, M. Budka, D. Hulbert, Bus journey simulation to develop public transport predictive algorithms, Soft Computing Letters 3 (2021) 100029.

[6] N. Van Oort, Service reliability and urban public transport design, Citeseer, 2011.

[7] J. G. Strathman, T. J. Kimpel, S. Callas, Headway deviation effects on bus passenger loads: Analysis of tri-met's archived avl-apc data, Tech. rep. (2003).

[8] A. Ceder, Public transit planning and operation: Theory, Modeling and practice. 0xford: Elsevier.

[9] A. Rudnicki, Measures of regularity and punctuality in public transport operation, IFAC Proceedings Volumes 30 (8) (1997) 661–666.

[10] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke, Combining machine learning and simulation to a hybrid modelling approach: Current and future directions, in: Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18, Springer, 2020, pp. 548–560.

[11] R. B. Noland, J. W. Polak, Travel time variability: a review of theoretical and empirical issues, Transport reviews 22 (1) (2002) 39–54.

[12] B. Büchel, F. Corman, Modelling probability distributions of public transport travel time components, in: 18th Swiss Transport Research Conference (STRC 2018), STRC, 2018.

[13] V. J. M. Low, H. L. Khoo, W. C. Khoo, Quantifying bus travel time variability and identifying spatial and temporal factors using burr distribution model, International Journal of Transportation Science and Technology 11 (3) (2022) 563–577.

[14] E. Mazloumi, G. Currie, G. Rose, Using gps data to gain insight into public transport travel time variability, Journal of transportation engineering 136 (7) (2010) 623–631.

[15] M. Taylor, Travel time variability—the case of two public modes, Transportation Science 16 (4) (1982) 507–521.

[16] E. B. Emam, H. Al-Deek, Using real-life dual-loop detector data to develop new methodology for estimating freeway travel time reliability, Transportation research record 1959 (1) (2006) 140–150.

[17] N. Uno, F. Kurauchi, H. Tamura, Y. Iida, Using bus probe data for analysis of travel time variability, Journal of Intelligent Transportation Systems 13 (1) (2009) 2–15.

[18] L.-M. Kieu, A. Bhaskar, E. Chung, Public transport travel-time variability definitions and monitoring, Journal of Transportation Engineering 141 (1) (2015) 04014068.

[19] E. Durán-Hormazábal, A. Tirachini, Estimation of travel time variability for cars, buses, metro and door-to-door public transport trips in santiago, chile, Research in Transportation Economics 59 (2016) 26–39.

[20] Z. Dai, X. Ma, X. Chen, Bus travel time modelling using gps probe and smart card data: A probabilistic approach considering link travel time and station dwell time, Journal of Intelligent Transportation Systems 23 (2) (2019) 175–190.

[21] A. Polus, A study of travel time and reliability on arterial routes, Transportation 8 (2) (1979) 141–151.

[22] W. C. Jordan, M. A. Turnquist, Zone scheduling of bus routes to improve service reliability, Transportation science 13 (3) (1979) 242–268.

[23] H. Al-Deek, E. B. Emam, New methodology for estimating reliability in transportation networks with degraded link capacities, Journal of intelligent transportation systems 10 (3) (2006) 117–129.

[24] S. Susilawati, M. A. Taylor, S. V. Somenahalli, Distributions of travel time variability on urban roads, Journal of Advanced Transportation 47 (8) (2013) 720–736.

[25] M. A. Taylor, Fosgerau's travel time reliability ratio and the burr distribution, Transportation Research Part B: Methodological 97 (2017) 50–63.

[26] M. Harsha, R. H. Mulangi, H. D. Kumar, Analysis of bus travel time variability using automatic vehicle location data, Transportation Research Procedia 48 (2020) 3283–3298.

[27] Z. Ma, L. Ferreira, M. Mesbah, S. Zhu, Modeling distributions of travel time variability for bus operations, Journal of Advanced Transportation 50 (1) (2016) 6–24.

[28] Z. Wall, D. Dailey, An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data, Master's thesis, Citeseer (1998).

[29] W.-H. Lin, J. Zeng, Experimental study of real-time bus arrival time prediction with gps data, Transportation Research Record 1666 (1) (1999) 101–109.

[30] F. Cathey, D. J. Dailey, A prescription for transit arrival/departure prediction using automatic vehicle location data, Transportation Research Part C: Emerging Technologies 11 (3-4) (2003) 241–264.

[31] A. Shalaby, A. Farhan, Bus travel time prediction model for dynamic operations control and passenger information systems, Transportation Research Board 2.

[32] R. Jeong, R. Rilett, Bus arrival time prediction using artificial neural network model, in: Proceedings. The 7th international IEEE conference on intelligent transportation systems (IEEE Cat. No. 04TH8749), IEEE, 2004, pp. 988–993.

[33] J. Patnaik, S. Chien, A. Bladikas, Estimation of bus arrival times using apc data, Journal of public transportation 7 (1) (2004) 1–20.

[34] L. Chu, S. Oh, W. Recker, Adaptive kalman filter based freeway travel time estimation, in: 84th TRB Annual Meeting, Washington DC, 2005.

[35] R. Padmanaban, L. Vanajakshi, S. C. Subramanian, Estimation of bus travel time incorporating dwell time for apts applications, in: 2009 IEEE Intelligent vehicles symposium, IEEE, 2009, pp. 955–959.

[36] B. A. Kumar, L. Vanajakshi, S. Subramanian, Pattern-based bus travel time prediction under heterogeneous traffic conditions, Transportation Research Record, Transportation Research Board, National Research Council, Washington, DC.

[37] S. I.-J. Chien, Y. Ding, C. Wei, Dynamic bus arrival time prediction with artificial neural networks, Journal of transportation engineering 128 (5) (2002) 429–438.
[38] M. Chen, X. Liu, J. Xia, S. I. Chien, A dynamic bus-arrival time prediction model based on apc data, Computer-Aided Civil and Infrastructure Engineering 19 (5) (2004) 364–376.
[39] B. Yu, W. H. Lam, M. L. Tam, Bus arrival time prediction at bus stop with multiple routes, Transportation Research Part C: Emerging Technologies 19 (6) (2011) 1157–1170.
[40] J. Pan, X. Dai, X. Xu, Y. Li, A self-learning algorithm for predicting bus arrival time based on historical data model, in: 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Vol. 3, IEEE, 2012, pp. 1112–1116.
[41] M. Yang, C. Chen, L. Wang, X. Yan, L. Zhou, Bus arrival time prediction using support vector machine with genetic algorithm, Neural Network World 26 (3) (2016) 205.
[42] T. Yin, G. Zhong, J. Zhang, S. He, B. Ran, A prediction model of bus arrival time at stops with multi-routes, Transportation research procedia 25 (2017) 4623–4636.
[43] J. Li, Bus arrival time prediction based on random forest, in: 2017 5th International Conference on Frontiers of Manufacturing Science and Measuring Technology (FMSMT 2017), Atlantis Press, 2017, pp. 867–872.
[44] X. Zhang, Z. Liu, Prediction of bus arrival time based on gps data: Taking no. 6 bus in huangdao district of qingdao city as an example, in: 2019 Chinese Control Conference (CCC), IEEE, 2019, pp. 8789–8794.
[45] C. Bai, Z.-R. Peng, Q.-C. Lu, J. Sun, Dynamic bus travel time prediction models on road with multiple bus routes, Computational intelligence and neuroscience 2015 (2015) 63–63.
[46] Z. Junyou, W. Fanyu, W. Shufeng, Application of support vector machine in bus travel time prediction, Int. J. Syst. Eng 2 (1) (2018) 21–25.
[47] P. He, G. Jiang, S.-K. Lam, D. Tang, Travel-time prediction of bus journey with multiple bus trips, IEEE Transactions on Intelligent Transportation Systems 20 (11) (2018) 4192–4205.
[48] B. Yu, H. Wang, W. Shan, B. Yao, Prediction of bus travel time using random forests based on near neighbors, Computer-Aided Civil and Infrastructure Engineering 33 (4) (2018) 333–350.
[49] B. A. Kumar, R. Jairam, S. S. Arkatkar, L. Vanajakshi, Real time bus travel time prediction using k-nn classifier, Transportation Letters 11 (7) (2019) 362–372.
[50] C. Chen, H. Wang, F. Yuan, H. Jia, B. Yao, Bus travel time prediction based on deep belief network with back-propagation, Neural Computing and Applications 32 (2020) 10435–10449.
[51] B. Ashwini, R. Sumathi, H. Sudhira, Bus travel time prediction: a comparative study of linear and non-linear machine learning models, in: Journal of Physics: Conference Series, Vol. 2161, IOP Publishing, 2022, p. 012053.
[52] F. Serin, Y. Alisan, M. Erturkler, Predicting bus travel time using machine learning methods with three-layer architecture, Measurement 198 (2022) 111403.
[53] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, Sumo–simulation of urban mobility: an overview, in: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation, ThinkMind, 2011.
[54] J. Hueper, G. Dervisoglu, A. Muralidharan, G. Gomes, R. Horowitz, P. Varaiya, Macroscopic modeling and simulation of freeway traffic flow, IFAC Proceedings Volumes 42 (15) (2009) 112–116.
[55] L. Smith, R. Beckman, K. Baggerly, Transims: Transportation analysis and simulation system, Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (1995).
[56] Q. I. Yang, H. N. Koutsopoulos, A microscopic traffic simulator for evaluation of dynamic traffic management systems, Transportation Research Part C: Emerging Technologies 4 (3) (1996) 113–129.
[57] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, Matsim-t: Architecture and simulation times, in: Multi-agent systems for traffic and transportation engineering, IGI Global, 2009, pp. 57–78.
[58] V. Leblond, L. Desbureaux, V. Bielecki, A new agent-based software for designing and optimizing emerging mobility services: Application to city of rennes, in: European Transport Conference 2020, 2020.
[59] G. Hamerly, E. Perelman, J. Lau, B. Calder, T. Sherwood, H. Hirsh, Using machine learning to guide architecture simulation., Journal of Machine Learning Research 7 (2).
[60] H. Shafizadeh-Moghadam, A. Asghari, A. Tayyebi, M. Taleai, Coupling machine learning, tree-based and statistical models with cellular automata to simulate urban growth, Computers, Environment and Urban Systems 64 (2017) 297–308.
[61] M. Elbattah, O. Molloy, B. P. Zeigler, Designing care pathways using simulation modeling and machine learning, in: 2018 Winter Simulation Conference (WSC), IEEE, 2018, pp. 1452–1463.
[62] S. Chabanet, H. B. El-Haouzi, P. Thomas, Coupling digital simulation and machine learning metamodel through an active learning approach in industry 4.0 context, Computers in Industry 133 (2021) 103529.
[63] H. Tongal, M. J. Booij, Simulation and forecasting of streamflows using machine learning models coupled with base flow separation, Journal of hydrology 564 (2018) 266–282.
[64] X. Yan, K. Xu, W. Feng, J. Chen, A rapid prediction model of urban flood inundation in a high-risk area coupling machine learning and numerical simulation approaches, International Journal of Disaster Risk Science 12 (6) (2021) 903–918.
[65] M. Shahhosseini, G. Hu, I. Huber, S. V. Archontoulis, Coupling machine learning and crop modeling improves crop yield prediction in the us corn belt, Scientific reports 11 (1) (2021) 1–15.
[66] H. Abdelaty, A. Al-Obaidi, M. Mohamed, H. E. Farag, Machine learning prediction models for battery-electric bus energy consumption in transit, Transportation Research Part D: Transport and Environment 96 (2021) 102868.
[67] R. E. Al Mamlook, A. Ali, R. A. Hasan, H. A. M. Kazim, Machine learning to predict the freeway traffic accidents-based driving simulation, in: 2019 IEEE National Aerospace and Electronics Conference (NAECON), IEEE, 2019, pp. 630–634.
[68] A. Sroczyński, A. Czyżewski, Road traffic can be predicted by machine learning equally effectively as by complex microscopic model, Scientific reports 13 (1) (2023) 14523.
[69] Z. Jiang, L. Shubin, L. Xiaoqing, Parameters calibration of traffic simulation model based on data mining, Journal of Transportation Systems Engineering and Information Technology 12 (6) (2012) 28–33.

[70] N. T. Ratrout, S. M. Rahman, I. Reza, Calibration of paramics model: Application of artificial intelligence-based approach, Arabian Journal for Science and Engineering 40 (2015) 3459–3468.
[71] I. Jang, D. Kim, D. Lee, Y. Son, An agent-based simulation modeling with deep reinforcement learning for smart traffic signal control, in: 2018 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2018, pp. 1028–1030.
[72] R. F. Daguano, L. R. Yoshioka, M. L. Netto, C. L. Marte, C. A. Isler, M. M. D. Santos, J. F. Justo, Automatic calibration of microscopic traffic simulation models using artificial neural networks, Sensors 23 (21) (2023) 8798.
[73] M. S. B. Othman, G. Tan, Machine learning aided simulation of public transport utilization, in: 2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), IEEE, 2018, pp. 1–2.
[74] B. Büchel, F. Corman, Review on statistical modeling of travel time variability for road-based public transport, Frontiers in Built Environment 6 (2020) 70.
[75] Google, General Transit Feed Specification, https://gtfs.org/, [Online; accessed 31-August-2023] (2023).
[76] M. M. Rahman, S. Wirasinghe, L. Kattan, Analysis of bus travel time distributions for varying horizons and real-time applications, Transportation Research Part C: Emerging Technologies 86 (2018) 453–466.
[77] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, J. Hu, Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation, Computational Materials Science 171 (2020) 109203.
[78] F. J. Massey Jr, The kolmogorov-smirnov test for goodness of fit, Journal of the American statistical Association 46 (253) (1951) 68–78.
[79] S. A. Arhin, E. C. Noel, O. Dairo, Bus stop on-time arrival performance and criteria in a dense urban area, International Journal of Traffic and Transportation Engineering 3 (6) (2014) 233–238.