# Improving deep-learning methods for area-based traffic demand prediction via hierarchical reconciliation

Mina Khalesian, Angelo Furno, Ludovic Leclercq

▶ **To cite this version:**

Mina Khalesian, Angelo Furno, Ludovic Leclercq. Improving deep-learning methods for area-based traffic demand prediction via hierarchical reconciliation. Transportation research. Part C, Emerging technologies, 2024, 159, pp.104410. 10.1016/j.trc.2023.104410 . hal-04701635

**HAL Id: hal-04701635**
**https://hal.science/hal-04701635v1**

Submitted on 23 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Manuscript File Click here to view linked References

# Improving deep-learning methods for area-based traffic demand prediction via hierarchical reconciliation

Mina Khalesian*, Angelo Furno, Ludovic Leclercq

*Univ Gustave Eiffel, ENTPE, LICIT-ECO7, Lyon, France*

## Abstract

Mobility services require accurate demand prediction both in space and time to effectively accomplish fleet rebalancing, present efficient on-demand transportation services, and allow for advanced ride-sharing with minimum fleet size. Although the optimization of mobility services is a widely studied topic, the critical demand prediction component has received less attention. In this paper, we aim to develop an efficient method for traffic demand forecasting by means of deep learning and hierarchical reconciliation approaches. The concepts, as well as the theories behind the proposed approach, are founded on a Hierarchical Time Series (HTS), which also adopts Long Short-term Memory (LSTM) as a special kind of Recurrent Neural Network (RNN) for the deep learning of the associated time series and producing reliable demand predictions. Herein, the proposed approach relies on the proper design of the HTS structure to find coherent forecasts for the number of trip departures and its associated uncertainty over predefined zones as well as over aggregated collections of these zones for applications like mobility service operations. Moreover, an error analysis is essential for accomplishing the reconciliation in the HTS structure optimally. The three main stages of the proposed approach (i.e., deep learning, error analysis, and optimal reconciliation), which independently function within the approach structure, have a remarkable ability to predict the demand, control all the forecasts at all levels of the hierarchical structure, and finally lead them to their coherent estimates. We evaluate the proposed approach on a large-scale GPS tracking dataset of Lyon in France. The proposed method reduces the root mean square error (RMSE) by 13.92% and 14.77% for the predefined and aggregated zones, respectively, compared with the LSTM using the historical demand and the external features of time at fifteen minutes time

* Corresponding author.
E-mail address: mina.khalesian@entpe.fr

resolution. Similarly, the corresponding improvement for mean absolute percentage error (MAPE) is 14.87% and 19.23%, respectively.

**Keywords:** Traffic Demand Prediction, Artificial Intelligence (AI), Deep Learning, LSTM, Hierarchical Time series, Hierarchical Reconciliation

# 1. Introduction

Traffic demand forecasting is one of the fundamental issues of any transportation system. Two kinds of problems have already received lots of attention: large-scale long-term demand prediction (e.g., forecasting typical daily OD matrix) and local short-term traffic state estimation (e.g., predicting the demand over the next hour). New mobility service operations impose new challenges for demand prediction, as we need high resolution both in space at large-scale and in time at short-term to mid-term (e.g., next fifteen minutes or next hours) to effectively perform fleet sizing and rebalancing.

Predicting traffic demands throughout a city can help car-sharing companies pre-allocate more cars in high-demand regions or help taxi centers to manage floating taxis by incentivizing vacant vehicles to move from the over-supply regions (the zones with more potential vehicles) to over-demand ones (the zones with more potential passengers) in advance. The same demand-supply imbalance exists in mobility-on-demand services such as e-hailing taxis, which have gained great popularity in recent years. Traffic demand forecasting can help to dispatch cars efficiently and consequently minimize the waiting time for both passengers and drivers (Li and Axhausen, 2019; Luo et al., 2021; Xu et al., 2017).

Traffic demand data varies with time and space and has complicated spatial-temporal dependencies. Regarding time dependency, the traffic demand is expected to be high during peak hours (morning and evening peaks) and low at night (sleeping hours). It also depends on the trend of the nearest historical demand. Furthermore, the traffic in each zone depends not only on the variables of that zone but also on all the other zonal variables in the whole area of interest, with a stronger impact from nearby zones than distant ones (Ke et al., 2017; Yang et al., 2010). The traffic sequence data is also affected by external features like weather conditions or events, time-of-day, and day-of-week. Although it is challenging to consider these exogenous factors and spatial and temporal dependencies simultaneously, it can help to improve the prediction.

Traffic demand prediction approaches can be divided into three categories. The first category is the statistical methods (Li et al., 2012; Moreira-Matias et al., 2013). Historical Average (HA), Auto-Regressive Integrated Moving Average (ARIMA), and Vector Auto-Regressive (VAR) are the most well-known statistical methods found in the literature. These algorithms are easy to be deployed but only applicable to relatively small datasets, and the capability of these approaches to deal with complex and dynamic traffic demand data is limited (Yin et al., 2021). Traditional machine learning methods (Guan et al., 2018; Li et al., 2018, 2017; Salinas et al., 2019)) constitute the second category. These methods, such as Support Vector Regression (SVR) and Random Forest Regression (RFR), can process high dimensional traffic data and capture non-linear relationships. However, with the advent of deep learning methods (Davis et al., 2020; Ke et al., 2017; Liu et al., 2019; Xu et al., 2017), which comprise the third category, the full potential of Artificial Intelligence (AI) has been utilized in traffic-related prediction applications (Nguyen et al., 2018). Several deep learning architectures, such as Convolutional Neural Network (CNN), Graph Convolutional Network (GCN) (Scarselli et al., 2008), Recurrent Neural Network (RNN) (Rumelhart et al., 1986), and its variants like Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014), have been used for traffic prediction, see (Yin et al., 2021) for a more extensive survey. In the following, we introduce some approaches that have used deep learning for traffic demand predictions.

Cheng et al. (2016) investigated a feature-level data fusion model to predict day-to-day travel demand variations based on the origin-destination matrices data of 30 consecutive days on a large-scale transportation network. They concluded that the proposed feature-level data fusion model, which integrated the feature attributes into long short-term memory (LSTM), outperformed the LSTM network and Deep Neural Network (DNN), which did not consider the external features.

Xu et al. (2017) proposed a sequence learning model based on LSTM for predicting taxi demands and showed that this approach outperformed the feed-forward neural network and naive statistic average predictor. In addition, Mixture Density Networks were added on top of LSTM to make the output as the parameters of a mixture distribution of the demand rather than directly forecasting the demand value. This probability distribution can be used to extract the predicted demand value for each area. They also investigated the effects of adding other features such as weather, time, and drop-offs and concluded that the models had close performances.

In (Ke et al., 2017), a deep learning (DL) approach named fusion convolutional long short-term memory network (FCL-Net) was employed by the fusion of convolutional techniques and LSTM network to predict short-term passenger demand for an on-demand transport service. Before feeding the explanatory variables into the DL structure, a tailored spatially aggregated random forest was used to rank these explanatory variables, and these ranks were used for feature selection. Based on the results, the authors concluded that the FCL-Net model provided better predictive performances compared to some statistical, machine learning, and deep-learning-based methods like HA, Moving Average (MA), ARIMA, Artificial Neural Network (ANN), LSTM, CNN, and eXtreme Gradient Boosting (XGBoost).

Li and Axhausen (2019) implemented six statistical, three machine learning (Random Forest, XGBoost, Multi-Layer Perceptron), and three deep learning models (LSTM-Neural Network, LSTM-onehot Encoding, and LSTM-Embedding) for short-term traffic demand prediction. They concluded that no specific model could get the best performance at all times and in the whole area. Based on their results, the six statistical methods have poor performances over the entire period and area. Machine learning methods and LSTM-Neural Network have better performance than other methods in terms of the SMAPE performance metric. LSTM-onehot Encoding and LSTM-Embedding show better performances in terms of RMSE. They compared the performance across all the unit areas and showed that LSTM-Neural Network performed best in most unit areas in their two datasets.

Luo et al. (2021) proposed a multi-task deep learning (MTDL) model based on LSTM to forecast short-term taxi demands at a multi-zone level. The proposed model was able to predict the demand of multiple zones simultaneously in a way that the demand prediction of each zone can be conducted by considering the information of zones that can help to improve the prediction. They concluded that their proposed method outperformed some benchmark algorithms like conventional LSTM, SVM, and k-nearest neighbors (k-NN).

Traffic demand characteristics change at different levels of temporal and spatial aggregations. In many traffic-related studies, authors mentioned that the aggregation level in the temporal dimension could affect the prediction accuracy so that aggregating in a longer time window results in smoother time series and therefore facilitates the pattern identification on a longer time horizon. However, aggregating in a longer time window may cause losing some important information at higher frequencies and remarkably diminish the accuracy. Therefore, finding a balance is noteworthy (Guo et al., 2007; Li et al., 2019; Oh et al., 2005; Vlahogianni and Karlaftis, 2011). Similarly, we can also consider the relationship between granularity and

accuracy in the spatial dimension. Generally, it seems that time series aggregated in more extensive areas are smoother, and then identifying their general trends is easier. Nonetheless, the time series will be noisier if small areas are used for aggregating traffic demands. However, in such a case, the data may contain useful information for forecasting the traffic demand in the future, especially for mobility service operations. Therefore, appropriate aggregation in the spatial and temporal dimensions can be used to improve prediction accuracy.

A hierarchical time series (HTS) is a collection of time series organized in a hierarchical aggregation structure. Forecast reconciliation is the process of improving the prediction accuracy by adjusting the forecasts to make them coherent across the hierarchy. This coherence can be checked at the spatial or temporal levels in time-series data related to traffic demand. For example, at the temporal level, the traffic demands at every hour of a day should add up coherently to give the diurnal traffic demand, or at the spatial level, the forecasts of traffic demand in initial traffic zones (here Traffic Analysis Zones (TAZ)) should add up to provide the forecasts of more extensive areas constructed based on initial traffic zones (here the zones designed to meet the requirements of mobility service operations).

In recent years, hierarchical time series and reconciliation have attracted much attention (Hollyman et al., 2021; Mancuso et al., 2021). However, aggregation in spatial and temporal dimensions hasn't been considered widely in the transportation field. Li et al. (Li et al., 2019) used a component-wise gradient boosting procedure (CWGB) combined with hierarchical reconciliation to predict traffic flow. They restricted their attention to linear base functions in one input argument only. In addition, since the correlation between locations varies significantly at different hours of a day, the boosting procedure is used separately on 24 hourly subsets to accommodate the spatial-temporal interactions in the model. They compared their approach with three frequently used methods, i.e., SARIMA, Kalman filter, and random forest, and concluded that the proposed approach outperformed these methods or performed at least as well as them when the information related to learning was limited.

In this study, an LSTM approach in combination with hierarchical reconciliation (HR) is proposed for short/mid-term forecasting of traffic demand. The proposed approach has three main features that make it different from the previous studies. First, the demand prediction for a specific area can be adjusted based on the demand forecasting of a group of regions at different aggregation levels (spatial levels) and the forecasts in different time resolutions (temporal levels). Second, the approach is based on a deep learning technique, LSTM

(Hochreiter and Schmidhuber, 1997), combined with an innovative application of HR. Third, the deep learning technique and the HR step are linked together via an error analysis, which can control the solutions to be in a true feasible space and can finally provide us with the expected precision of the forecasted demands.

The rest of this paper is organized as follows. First, Section 2 describes the methodology of the proposed approach in detail, including problem definition, hierarchical time series structure, bottom-up approach, top-down approach, deep learning, optimal reconciliation approach, error analysis, and method structure. Subsequently, Section 3 is dedicated to numerical experiments, which evaluate the performance of the proposed approach for demand prediction in the hierarchical time series structure of zones based on the real-world dataset of the floating car in Lyon. Finally, Section 4 concludes the main findings of the research.

# 2. Methodology

In the current research, we consider two specific spatial partitioning levels: the initial one consists of homogeneously populated areas based on census data, and the second one is the aggregation of these zones into larger areas that still fulfill a maximum service time criterion for mobility services. In short, those areas should be large enough to allow robust demand predictions but small enough to allow pre-positioned vehicles within the area to serve any internal requests with a low waiting time for passengers. In the proposed method, apart from adopting a deep learning technique for demand prediction, the main idea is that all the predicted demands in all zones and regions of the mentioned partitioning levels should be adjusted and matched with each other in the hierarchical tree structure.

## 2.1. Problem Definition

The traffic demand prediction problem can be defined as forecasting the number of trips initiated (corresponding to potential requests for mobility service) at a future time interval in a specific region based on the given historical requesting data (historical traffic demands) as well as other features such as time feature (i.e., time-of-day, and day-of-week) and spatial feature (i.e., the area identifier). Generally, this problem can be considered a time series forecasting problem.

To forecast the demands accurately, we need to divide the city (or the area of interest) into regions and predict the demand based on those regions. In Appendix A, a method for

aggregating predefined zones to zones of adequate size for mobility service operations is proposed. On the other hand, we should consider a predefined time interval in which the variables (such as demand intensity) be aggregated. For instance, in the numerical experiments of the current research, intervals with a length of fifteen minutes and one hour have been considered for this purpose.

## 2.2. Hierarchical Time Series structure

The collection of time series organized in a hierarchical aggregation structure is called Hierarchical Time Series (HTS). In HTS, aggregate forecasts can be generated by adding up the disaggregate ones in the same way as the data in the hierarchical aggregation structure. Nonetheless, the challenge is to be sure that the predictions are coherent at different aggregation levels and add up the forecasts in a way that is consistent with the aggregation structure. Therefore, the forecasts at different hierarchy levels must be adjusted (in other words, reconciled) regardless of the methods used for initially forecasting the time series. In the following, we provide the basic concepts related to HTS, while in Section 2.6, the optimal reconciliation will be discussed.

First of all, consider a two-level hierarchical structure in which level 0 (i.e., "Total") is the most aggregate level of the data, and the 2nd level (the bottom level) contains the most disaggregate time series (Fig. 1). Moreover, consider that the total desired area and the level of the hierarchical structure are constituted by two sets of zones $\{z^i\}$ and $\{z^i_j\}$ that form the 1st (top) and 2nd (bottom) levels of the HTS structure (see Fig. 1).

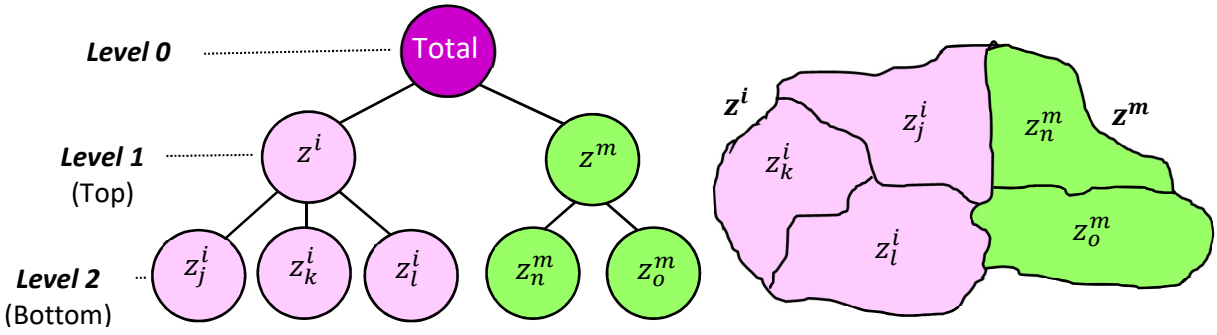


**Fig. 1.** Left) a presumed two-level hierarchical tree structure for the zones, and Right) a schematic figure of a two-level structure in the spatial domain

Hence, in such a two-level hierarchical structure, the zone set $\left\{z_j^i\right\}$ is supposed to be constituted by subsets like $\left\{z_1^i, z_2^i, \ldots, z_{N_i}^i\right\}$ (i.e., $\left\{z_j^i\right\} = \bigcup_{i=1}^{N^{\mathrm{Up}}} \left\{z_1^i, z_2^i, \ldots, z_{N_i}^i\right\}$), and the zones belonging to the subset $\left\{z_1^i, z_2^i, \ldots, z_{N_i}^i\right\}$ are aggregated together to create the zone $z^i$ as follows:

$$z^i = \bigcup_{j=1}^{N_i} z_j^i \tag{1}$$

where the zones $\left\{z_j^i\right\}$ and $\left\{z^i\right\}$ constitute the bottom and top levels of the two-level hierarchical structure, respectively; $N^{\mathrm{Up}}$ is the number of the aggregated zones at the top level; also, the number of all disaggregated zones at the bottom level $N_{\mathrm{Down}}$ is equal to $\sum_{i=1}^{N^{\mathrm{Up}}} N_i$, and $N_i$ is the number of zones at the bottom level connected to the zone $z^i$ as the *i*$^{th}$ aggregated zone at the top level.

**Definition 1 (Conservation Equation):** the demands of all zones $\left\{z^i\right\}$ and $\left\{z_j^i\right\}$ at time $t$ should satisfy the following equation, herein called Conservation Equation, which means that the quantity of demand should be conserved at all levels of the hierarchical structure:

$$d^i\left(t\right) = \sum_{j=1}^{N_i} d_j^i\left(t\right) \tag{2}$$

where $d^i\left(t\right)$ and $d_j^i\left(t\right)$ are the demands at time $t$ in zones $z^i$ and $z_j^i$, respectively, and this equation of demands is the basic foundation of the two-level hierarchical approach. It is worth mentioning that although the current study does not consider any overlap or intersection between the zones, in other studies, based on their requirements and applications, it may be expected to have zones with overlaps and intersections. In such a case, based on the inclusion-exclusion principle in set theory, we should expect other additional terms to appear on the right side of the conservation equation.

Accordingly, the conservation equation between the demands (i.e., Eq.(2)) will imply the following equation between the top and bottom levels of the hierarchical structure.

$$\mathbf{d} = \mathbf{S}\mathbf{d}_{\mathrm{Down}} \tag{3}$$

where the matrix $\mathbf{S}$ is called the summing matrix, the vector $\mathbf{d}$ includes the demands in all the levels, and $\mathbf{d}_{\mathrm{Down}}$ is comprised of the demands in the bottom level. For instance, for the schematic hierarchical tree structure illustrated in Fig. 1, the above equation, as well as the corresponding matrices, will be as follows:

$$\underbrace{\begin{bmatrix} \vdots \\ d^{i} \\ d^{m} \\ d^{i}_{j} \\ d^{i}_{k} \\ d^{i}_{l} \\ d^{m}_{n} \\ d^{m}_{o} \\ \vdots \end{bmatrix}}_{\mathbf{d}\ (N_{\text{Down}}+N^{\text{Up}})\times 1} = \underbrace{\begin{bmatrix} \ddots & & & \vdots & & & \iddots \\ & 1 & 1 & 1 & 0 & 0 & \\ & 0 & 0 & 0 & 1 & 1 & \\ & 1 & 0 & 0 & 0 & 0 & \\ \cdots & 0 & 1 & 0 & 0 & 0 & \cdots \\ & 0 & 0 & 1 & 0 & 0 & \\ & 0 & 0 & 0 & 1 & 0 & \\ & 0 & 0 & 0 & 0 & 1 & \\ \iddots & & & \vdots & & & \ddots \end{bmatrix}}_{\mathbf{S}\ (N_{\text{Down}}+N^{\text{Up}})\times N_{\text{Down}}} \underbrace{\begin{bmatrix} \vdots \\ d^{i}_{j} \\ d^{i}_{k} \\ d^{i}_{l} \\ d^{m}_{n} \\ d^{m}_{o} \\ \vdots \end{bmatrix}}_{\mathbf{d}_{\text{Down}}\ N_{\text{Down}}\times 1} \tag{4}$$

The definitions above can be easily generalized to any N-level hierarchy of non-overlapping zones.

## 2.3. Bottom-up approach

The above relations can be used for the bottom-up approach (e.g., see (Athanasopoulos et al., 2009; Hyndman et al., 2011)). In this case, we predict the demands in all zones and all the levels via forecasting the demands at the bottom level of the hierarchical structure (the most disaggregate level of the data), so no information will be lost due to the aggregation. Nevertheless, the bottom-level data can be noisy, and this may result in reduced overall forecast accuracy. It is worth mentioning that the predictions are usually more robust at the top level as total demand is higher with fewer spatial variations. Therefore, the main disadvantage of this approach is that it completely ignores any information about the predictions in the top levels, while we need to have a coherent prediction for all the zones, which considers all the gathered information and initial demands forecasts.

## 2.4. Top-down approach

In the top-down approach, first, the base forecasts for the top-level time series are generated, and then only these predictions are disaggregated down the hierarchy to generate the forecasts for the bottom-level time series. In this regard, a set of disaggregation proportions is required that represents the relative contribution of the bottom-level time series to the time series at the top levels of the hierarchical structure. The generated bottom-level predictions are aggregated to make other time series forecasts in the hierarchical structure. In general, the top-down approach can be represented as follows (e.g., see (Athanasopoulos et al., 2009; Hyndman et al., 2011)):

$$\hat{\mathbf{d}} = \mathbf{S}\mathbf{P}^{\text{Up}}_{\text{Down}}\mathbf{d}^{\text{Up}} \tag{5}$$

where $\mathbf{S}$ is the summing matrix, $\mathbf{P}_{\text{Down}}^{\text{Up}}$ is a proportion matrix between the top and bottom levels, which maps the vector of demands $\mathbf{d}^{\text{Up}}$ at the top-level to the vector of demands at the bottom-level, and $\hat{\mathbf{d}}$ is the vector of coherent forecasts of demands at all levels.

The Average Historical Proportions (AHP) and the Proportions of the Historical Averages (PHA) are the two most common top-down approaches, which are based on the proportions of the historical data (Hyndman and Athanasopoulos, 2018). These approaches seem to generate reliable forecasts for the aggregate levels, and therefore they are useful when we have a low amount of data. On the other hand, we may have a lack of information due to the aggregation and be unable to capture and take advantage of individual time series characteristics such as time dynamics and therefore have a less accurate prediction at the bottom-level of the hierarchy. The approaches based on historical data generate less accurate forecasts at the bottom-level of the hierarchy because they do not consider how the proportions change over time (Mancuso et al., 2021). Now, before exploiting the fundamental equations of the HTS described in bottom-up and top-down approaches (i.e., Eqs. (3) and (5)), we need to provide the HTS with initial forecasts at each level of the hierarchical structure.

## 2.5. Deep learning

In this research, the initial prediction of demands is performed based on Long Short-term Memory (LSTM) as a special kind of Recurrent Neural Network (RNN) (Hochreiter and Schmidhuber, 1997). Contrary to traditional Artificial Neural Networks (ANNs) that are not able to handle the sequential nature of traffic demand data and, in fact, are not able to take into consideration the time series characteristics such as temporal dependencies (Luo et al., 2021), Recurrent Neural Network (RNN) (Connor et al., 1994) is one of the most popular models that can process time series data and overcome this shortcoming by considering the connection between units by timestamps. The idea behind RNN is to use a memory in which some relevant parts of the past input data are stored and used while predicting the output data in the future. However, since RNNs cannot properly fit the time series with long-time lags because of vanishing and exploding gradients, the LSTM, as a variant of RNN, is capable of learning long-term dependencies by remembering information for long periods (Hochreiter and Schmidhuber, 1997).

Each LSTM cell possesses three kinds of gates that allow it to store and update the information over a long period (Hochreiter and Schmidhuber, 1997): 1) Forget gate ( $f$ ): decides what kind

of information will be thrown away from the cell state, 2) Input gate ($i$): decides the value we need to update, in other words, what new information is going to be stored in the cell state (a sigmoid layer), and 3) Output gate ($o$): After updating the old cell state, it decides what information we are going to output to the next cell based on the cell state. Thus, the core idea of the LSTM structure employed by the current research can be summarized by the following equations (Hochreiter and Schmidhuber, 1997):

$$\begin{cases} f_n = \sigma\left(W_{df} d_n + W_{hf} h_{n-1} + b_f\right) \\ i_n = \sigma\left(W_{di} d_n + W_{hi} h_{n-1} + b_i\right) \\ o_n = \sigma\left(W_{do} d_n + W_{ho} h_{n-1} + W_{co} \circ c_n + b_o\right) \\ c_n = f_n \circ c_{n-1} + i_n \circ \tanh\left(W_{dc} d_n + W_{hc} h_{n-1} + b_c\right) \\ h_n = o_n \circ \tanh\left(c_n\right) \end{cases} \tag{6}$$

where the input is a training vector sequence of demand $\mathbf{d}_{\text{Train}} = \left[d_n\right]_{n=1,\ldots,N_{\text{Train}}}$ called training dataset; the output is a hidden vector sequence $\mathbf{h} = \left[h_n\right]_{n=1,\ldots,N_{\text{Train}}}$; $W$ indicates the weight matrix; $b$ denotes the bias; $c$ represents the memory cell; the operator $\circ$ refers to an element-wise vector product (Hadamard product), and the functions $\sigma$ and $\tanh$ are the sigmoid and tangent activation functions, respectively (i.e., $\sigma(x) = (1 + e^{-x})^{-1}$ and $\tanh(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$).

Accordingly, based on the abovementioned LSTM deep learning architecture, if the training vector sequence of demand is considered as $\left[d_n = d(t - n\Delta t)\right]_{n=1,\ldots,N_{\text{Train}}}$, a preliminary prediction of the demand at time $t$ will be the last output of the deep learning structure, and it can be obtained as follows:

$$d(t) = \mathbf{LSTM}\left(\left[d(t - n\Delta t)\right]_{n=1,\ldots,N_{\text{Train}}}\right) \tag{7}$$

where **LSTM** denotes the LSTM deep learning network, $N_{\text{Train}}$ is the number of timeslots in the training set, and $\Delta t$ is the length of timeslots.

Accordingly, now the initial values of the demands, predicted via the deep learning approach, are ready to be used for finding a reconciliation at hierarchical structure levels.

## 2.6. Optimal reconciliation approach

Overall, according to the HTS approaches discussed so far (see Eqs. (3) and (5)), the corresponding equation of the HTS reconciliation can be generalized as follows:

$$\hat{\mathbf{d}} = \mathbf{R}\mathbf{d}$$
$$\mathbf{R} = \mathbf{S}\mathbf{P} \tag{8}$$

In the above relation, the vector $\mathbf{d}$ includes the initial forecasts of all zones as the base forecasts, independently obtained by the LSTM without considering the aggregation constraints (i.e., at time $t$ we have $\mathbf{d}(t) = \mathbf{LSTM}(\left[\mathbf{d}(t - n\Delta t)\right]_{n=1,\dots,N_{\text{Train}}})$), and it is stacked in the same order as the coherent forecasts $\hat{\mathbf{d}}$ (Eq.(4) shows the order in the vector $\mathbf{d}$ including all the demands). The matrix $\mathbf{R}$ denotes a "Reconciliation matrix" that reconciles the base forecasts $\mathbf{d}$ to generate coherent ones $\hat{\mathbf{d}}$. Also, $\mathbf{P}$ is a matrix that maps the preliminary forecasts of all zones $\mathbf{d}$ into the bottom-level (like the matrix $\mathbf{P}_{\text{Down}}^{\text{Up}}$ in the top-down approach that maps the vector of demands at the top-level to the vector of demands at the bottom-level), and $\mathbf{S}$ is the summing matrix defined by Eqs. (3) and (4).

In this regard, ideally, we need to find the matrix $\mathbf{P}$ that minimizes the error of coherent forecasts. Wickramasuriya et al. (Wickramasuriya et al., 2019) showed that the optimal version of $\mathbf{P}$ can be given by:

$$\mathbf{P} = (\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}}\mathbf{S})^{+}\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}} \tag{9}$$

where the matrix $\mathbf{W}_{\mathrm{d}}$ is the weight matrix associated with the initial predicted demand $\mathbf{d}$.

Therefore, considering Eq.(8), the optimal reconciled forecast, which is referred to as the "Minimum Trace" estimator or "MinT", is presented as follows:

$$\hat{\mathbf{d}} = \mathbf{S}(\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}}\mathbf{S})^{+}\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathrm{d}}\mathbf{d} \tag{10}$$

where the sign (+) denotes the Moore–Penrose pseudo-inverse, and the weight matrix $\mathbf{W}_{\mathrm{d}}$ is equal to the Moore–Penrose inverse of the variance-covariance matrix $\mathbf{C}_{\mathrm{d}}$ associated with the base forecast errors of $\mathbf{d}$ (i.e., $\mathbf{W}_{\mathrm{d}} = \mathbf{C}_{\mathrm{d}}^{+}$).

## 2.7. Error Analysis

In practice, for implementing the idea of MinT and the above-mentioned equation of the reconciliation (i.e., Eq.(10)), the matrices $\mathbf{C}_{d}$ and $\mathbf{W}_{d}$ are required to be optimally estimated. For this purpose, some approximations are proposed in the literature, such as (Hyndman and

Athanasopoulos, 2018) and (Wickramasuriya et al., 2019). Continuing such studies, herein, we first aim to find the behavior of the random errors occurring at the initial forecasts of the demands obtained by the LSTM and then compute the variance-covariance matrix required by the reconciliation. In this regard, we consider the data in three parts: training, validation, and test data. After training the deep learning network and obtaining the initial forecasts using the training data, the validation part is utilized to analyze random errors occurring at the initial forecasts based on the true values of the demands.

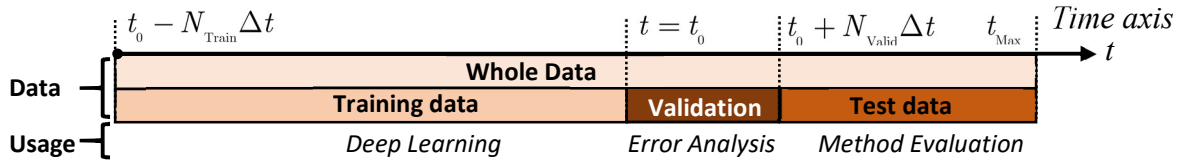


**Fig. 2.** The suggested division of the data used by the proposed method

This validation data as well as its error analysis can then enable us to predict the main behavior of the random errors caused by LSTM computations and estimate the variance-covariance matrix $\mathbf{C}_d$ and the weight matrix $\mathbf{W}_d$ associated with the base forecast errors, required by the reconciliation (see Eq.(10)). Finally, in the test part of the data, first, we assume there is no knowledge of the true values of demands. So, we can run the reconciliation procedures and then verify the results via the true values of demands known by the test data. Fig. 2 summarizes the suggested division of the data into three parts and its usage in the proposed method.

Hence, based on the error analysis, the variance-covariance matrix $\mathbf{C}_{\text{d}}$ as well as the weight matrix $\mathbf{W}_{\text{d}}$ can be estimated as follows (e.g., see (Hyndman and Athanasopoulos, 2018) and (Wickramasuriya et al., 2019)):

$$\begin{aligned} \mathbf{C}_{\text{d}} &= diag(\mathbf{var}_{\text{d}}) \\ \mathbf{W}_{\text{d}} &= \mathbf{C}_{\text{d}}^{+} \end{aligned} \tag{11}$$

where the variance-covariance matrix $\mathbf{C}_{\text{d}}$ is considered to be a diagonal matrix whose diagonal elements are the elements of the vector $\mathbf{var}_{\text{d}}$ including the variances of the initial forecasts (i.e., $\mathbf{var}_{\text{d}} = 1/N_{\text{Valid}} \sum_{n=1}^{N_{\text{Valid}}} \mathbf{v}_n \circ \mathbf{v}_n$); $\circ$ refers to the element-wise vector product (Hadamard product); $\mathbf{v}_n \in \mathbb{R}^{N_{\text{Zones}}}$ is an estimate of the error vector of the $n^{\text{th}}$ base forecast in the validation data, which is obtained by the true value and the initial forecast of the demand in the validation data set; and $N_{\text{Valid}}$ and $N_{\text{Zones}}$ are the number of validation data and the number of zones, respectively (i.e., $N_{\text{Zones}} = N_{\text{Down}} + N^{\text{Up}}$).

The optimal reconciliation approach based on such variance-covariance and weight matrix is usually referred to as the Weighted Least Square (WLS), in which the base forecasts are scaled by the inverse of the variances. Accordingly, if we have an accurate estimation of the variance-covariance of the base forecasts, which can be obtained by Eq.(11), the variance-covariance of the coherent demand forecasts $\hat{\mathbf{d}}$ can also be estimated by using the properties of the pseudo-inverse and the uncertainty propagation (variance-covariance propagation) in the MinT equation (Eq.(10)) as follows (*c.f.* (Björck, 1996; Mikhail, 1982)):

$$\mathbf{C}_{\hat{\mathbf{d}}} = \mathbf{S}(\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathbf{d}}\mathbf{S})^{+}\mathbf{S}^{\mathrm{T}} \tag{12}$$

where the diagonal elements of the variance-covariance $\mathbf{C}_{\hat{\mathbf{d}}}$ associated with the predicted vector present us with the uncertainty and precision of the predicted demands in all zones.

### 2.7.1. Filtering

As we mentioned before, the main advantage of the validation data is that both the initial forecasts and their true values are known in this part of the data. This enables us to have an accurate insight into the feasible space of the demand predictions. After forming the feasible space by the known values of the true demands, projecting the initial forecasts onto it can prevent the following forecasts from diverging. In other words, the predictions can also be controlled by means of such a feasible space of forecasts, which is constructed by the known true values of the demands in the validation data. Moreover, having such reliable space can also provide us with confident filtering of the initial data and present us with an accurate estimation of the error vector $\mathbf{v}$ required for estimating the variance-covariance matrix $\mathbf{C}_{\mathbf{d}}$ (see Eq.(11)).

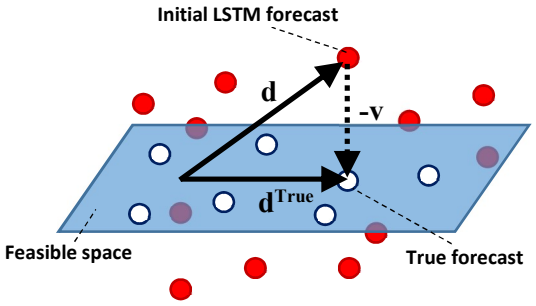


**Fig. 3.** The projection of the initial forecast on the feasible space formed by true values of the forecasts

Hence, consider a feasible space made by the demands' true values, accompanied by the initial forecasts occurring randomly around such space (see Fig. 3). Now, based on the least square method, if we have some information about this space (like the true values of the demands in

the validation part), at each step, we can project the initial solution $\mathbf{d}$ on this space and find the optimum solution in the feasible space. In fact, such projection of the forecasts on the true feasible space of the solutions can also be counted as a filtering of the initial predictions, where we can also find an estimation of the error vector $\mathbf{v}$. For this purpose, suppose that the validation data is comprised of $N_{\text{Valid}}$ epochs such as $\left\{\mathbf{d}_n = \mathbf{d}(t = t_0 + n\Delta t) \in \mathbb{R}^{N_{\text{Zones}}}\right\}_{n=1,2,..,N_{\text{Valid}}}$, and also, suppose that the true values of the demand vectors in the validation data $\left\{\mathbf{d}_n^{\text{True}}\right\}_{n=1,2,..,N_{\text{Valid}}}$ are known and constitute the columns of the matrix $\mathbf{D} = \left[\mathbf{d}_1^{\text{True}} \quad \mathbf{d}_2^{\text{True}} \quad \cdots \quad \mathbf{d}_{N_{\text{Valid}}}^{\text{True}}\right]$. Now, based on the least square method, which aims to minimize the vector $\mathbf{v}$ (i.e., $\left\|\mathbf{v}\right\|_2^2 \to Min$) (e.g., see (Björck, 1996)), the following projection matrix $\mathbf{F}$ is able to project the initial forecast $\mathbf{d}$ on the feasible space as follows (e.g., see (Banerjee and Roy, 2014)):

$$\begin{aligned} &\mathbf{F} = \mathbf{D}(\mathbf{D}^{\mathrm{T}}\mathbf{D})^{+}\mathbf{D}^{\mathrm{T}} \\ &\tilde{\mathbf{d}} = \mathbf{F}\mathbf{d} \end{aligned} \tag{13}$$

where $\tilde{\mathbf{d}}$ is the projection of the initial demand $\mathbf{d}$ on the feasible space, and it is an estimation of the true solution $\mathbf{d}^{\text{True}}$. According to the above equation, we can easily realize how the matrix $\mathbf{F}$ possesses the main property of projection matrices and satisfies $\mathbf{F} = \mathbf{F}\mathbf{F}$.

In fact, Eq.(13) applies filtering to the initial demand $\mathbf{d}$, and $\tilde{\mathbf{d}} = \mathbf{F}\mathbf{d}$ is the filtered version of the demand vector $\mathbf{d}$. Consequently, we can have an estimation of the error vector $\mathbf{v}$ as follows:

$$\mathbf{v} = (\mathbf{I} - \mathbf{F})\mathbf{d} = (\mathbf{I} - \mathbf{D}(\mathbf{D}^{\mathrm{T}}\mathbf{D})^{+}\mathbf{D}^{\mathrm{T}})\mathbf{d} \tag{14}$$

where $\mathbf{I} \in \mathbb{R}^{N_{\text{Zones}}}$ is an identity matrix. Herein, this evaluation of the error is used for estimating the required variance-covariance matrix defined by Eq.(11) where $\mathbf{v}_n \in \mathbb{R}^{N_{\text{Zones}}}$, the estimate of the error in the $n^{\text{th}}$ base forecast of the validation data, is obtained via Eq.(14) (i.e., $\mathbf{v}_n = (\mathbf{I} - \mathbf{F})\mathbf{d}_n$).

Accordingly, considering the deep learning and the optimal reconciliation as well as the error analysis described in this section, the final adapted relation for the optimal reconciliation and demand forecasting of all the hierarchical levels at time $t$ can be obtained as follows (see (*c.f.* Eqs. (7), (10), and (13))):

$$\hat{\mathbf{d}}(t) = \mathbf{S}(\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathbf{d}}\mathbf{S})^{+}\mathbf{S}^{\mathrm{T}}\mathbf{W}_{\mathbf{d}}\mathbf{F}(\mathbf{LSTM}(\left[\mathbf{d}(t - n\Delta t)\right]_{n=1,\ldots,N_{\text{Train}}})) \tag{15}$$

where the summing matrix $\mathbf{S}$, the weight matrix $\mathbf{W}_d$, and the filtering matrix $\mathbf{F}$ are obtained by Equations (4), (11) and (13), respectively.

## 2.8. Method structure

According to the above-motioned steps, the following diagram illustrated in Fig. 4 can also summarize the structure of the proposed method. The structure starts with the training data set at time $t = t_0$ and ends with the coherent reconciled demand forecasts at time $t = t_{\text{Max}}$. The intervals $T_{\text{Train}} = \left[t_0 - N_{\text{Train}}\Delta t, t_0\right[$, $T_{\text{Valid}} = \left[t_0, t_0 + N_{\text{Valid}}\Delta t\right]$, and $T_{\text{Test}} = \left]t_0 + N_{\text{Valid}}\Delta t, t_{\text{Max}}\right]$ represent the time intervals associated with the training, validation, and test data sets, respectively.

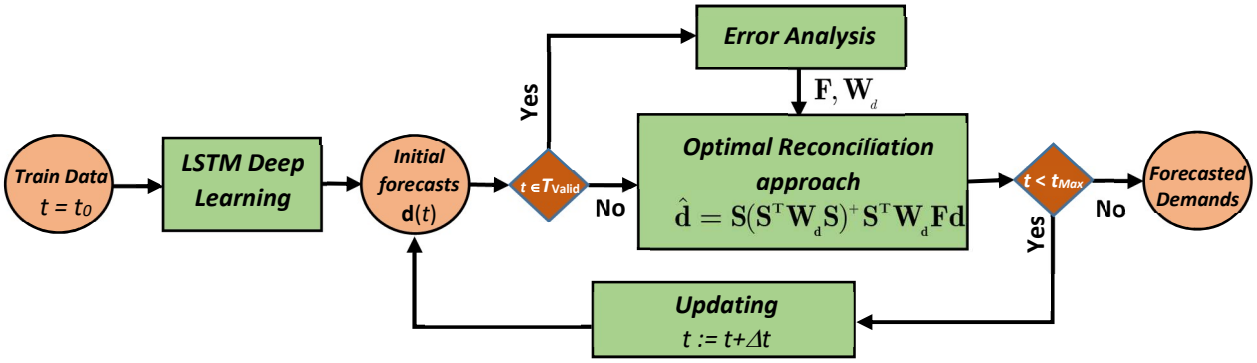


**Fig. 4.** The structure of the proposed method

According to the methodology, the proposed approach is comprised of three main stages. The aims and ideas of these stages can be summed up as follows:

1- Deep learning: This step of the proposed approach can present us with initial predictions of the demands while we know that these initial forecasts could be inaccurate. But its main advantage is that this step can provide us with a correct trained model of the demands when there is no knowledge about the behavior of the demand time series and the stochastic behavior of the demand with respect to the time is hard to model. In this step, utilizing other information about the demands as additional input (e.g., external features of time) can be beneficial to network training.

2- Error analysis: The main idea behind this step is to validate and control the initial forecasts of deep learning. The initial forecasts released from the first step (i.e., Deep learning) are not ready to be used by the hierarchical reconciliation and their random

errors need to be investigated and controlled (see Section 2.7). One of the main outputs of the error analysis is the weight matrix, which is essentially required by optimal reconciliation. Meanwhile, the error analysis step possesses another main output, a projection matrix, which provides the proposed approach with a filtering. By means of the projection to the feasible space of solutions, the filtering avoids the initial forecasted time series diverging from the true solutions (see Section 2.7.1).

3- Optimal reconciliation: the main idea behind this step is to optimally adjust and match all the demands with the hierarchical tree structure of zones (see Fig. 1). This step enables the proposed approach to present coherent forecasts of the demands considering all the conservation equations of the demands in all levels of the Hierarchical structure.

# 3. Numerical experiments

In this section, we evaluate the effectiveness of our proposed approach on the defined dataset and survey how well it can predict the demands in the future by comparing it with the initial deep learning and hierarchical methods described in Sections 2.3, 2.4, 2.5, and 2.6. In this regard, we aim to separately assess the ideas of the stages that construct the proposed approach. Accordingly, the numerical experiments of the research are organized and classified into the following parts:

1- Assessing the effect of external features of time on the deep learning stage when the time features are considered as additional inputs of the deep learning network.
2- Assessing the effectiveness of using both the optimal reconciliation and error analysis parts of the proposed method by comparing the forecasts obtained by the proposed approach with the LSTM forecasts considering the external features of time.
3- Assessing the necessity of the hierarchical reconciliation step for enhancing the accuracy of demand prediction by comparing the results of the proposed method with the ones obtained by the bottom-up approach.
4- Assessing the role of an accurate weight matrix in the optimal reconciliation via comparing the forecasts obtained by the proposed method (using the weight matrix approximated by the error analysis) with an Ordinary Least Square (OLS).
5- Assessing the role of filtering in the proposed approach via comparing the reconciliation (without filtering) with the reconciliation accompanied by the filtering in the proposed approach.

In the following section, details on the data set are presented. Then we go through the details of the numerical experiments associated with the above-mentioned assessments of the proposed approach.

## 3.1. Data

In our work, we use a floating car dataset (GPS trajectory measurements) provided by the Be-Mobile group[1] from 01/04/2021 to 01/31/2021 in urban and peripheral areas in Lyon, which is from $4.576^{\circ}\ E$ to $5.222^{\circ}\ E$ in longitude and from $45.474^{\circ}\ N$ to $46.001^{\circ}\ N$ in latitude. This dataset includes the sample trajectory points of the floating cars. The sampling rate (the average number of samples obtained in one second) is about 0.5-1HZ (i.e., 1 sample per 1-2 seconds). Each record provides a segment ID (the longitude and latitude of the start and the end of the segments can be extracted from the road network), Vehicle ID, Provider ID, Travel time, and GPS timestamp.

The road network consists of 287,888 road segments characterized by different road attributes, such as the length of segments, the average speed in a non-congested situation (the free-flow speed), and the longitude and latitude of the start and end points. The study area consists of 679 TAZ zones (the IRIS[2] zones), and the shape file of these zones is co-published by INSEE and IGN (2021). The number of trips that start in each zone can be counted during each time-step length (the time interval), and the historical data of demand intensity in each zone can be calculated by aggregating the number of trips started from that zone during each time interval. These data sequences (the time series) are fed into the LSTM for sequential pattern learning. Herein, in order to avoid using future information in the learning procedure and have a careful validation of the predictions as well, the dataset, which includes one-month data (four weeks) starting from the first business week of 2021 (i.e., January 4th) and ending at the beginning of February 2021 (i.e., January 31st), is divided into three subsets: 1) the training set comprised of observations (acquired in the first three weeks), 2) the validation set (consisting of 60 samples acquired after the training data), and 3) the test set comprised of the remaining observations.

[1] https://be-mobile.com
[2] Ilots Regroupés pour l'Information Statistique

## 3.2. Aggregation of zones

As mentioned before, we consider two particular partitioning levels in this paper. The first one is the specific spatial zoning called IRIS zones that are considered as Traffic Analysis Zone (TAZ) data which divides our area of interest into 679 zones. IRIS zoning is the spatial subdividing of an urban area based on the census data of its residents. IRIS project almost considered the same number of inhabitants in each subarea (about 2000), where the shapes and surfaces of the areas are highly variable. In the IRIS zones, the smallest size may include two to three streets, and the largest may cover several square kilometers.

The advantage of using IRIS zones is that it is a well-known partitioning in which the population is homogenous. Hence, the demand predictions within the zones are comparable regarding their similar population ranges. The problem is that these zones are small, and the data can be noisy. So, we need to define another partitioning. Herein, the idea is to aggregate the IRIS zones to have bigger ones that are still adequate to run a mobility service. In this regard, the second partitioning level is a set of aggregations of IRIS zones that is consistent with the requirement of the mobility service operations.

Therefore, in practice, we want to aggregate the IRIS zones based on the spatial information of the zones and some constraints on service time. In this regard, we performed the aggregation of the zones to have a new set of zones with an average travel time of 5 to 10 minutes in each zone. This time threshold is considered a requirement of the mobility service operations as the zones' sizes are meaningful in terms of mobility service operations, and the service providers can manage the drivers in that zone to reach the passengers within an acceptable time interval.

In Appendix A, you can find the details on how we performed the aggregation of the IRIS zones. The new zones divide the area of interest into 139 regions. Figures 5 and 6 illustrate the spatial distributions and histograms associated with the average demand per quarter-hour for all the old zones (the predefined IRIS zones) and new zones (the aggregated zones) in January 2021. These results were obtained via the Be-Mobile group's floating car dataset. Due to the aggregation, the number of demands has increased in the new zones compared to the old zones (see Figures 5 and 6).

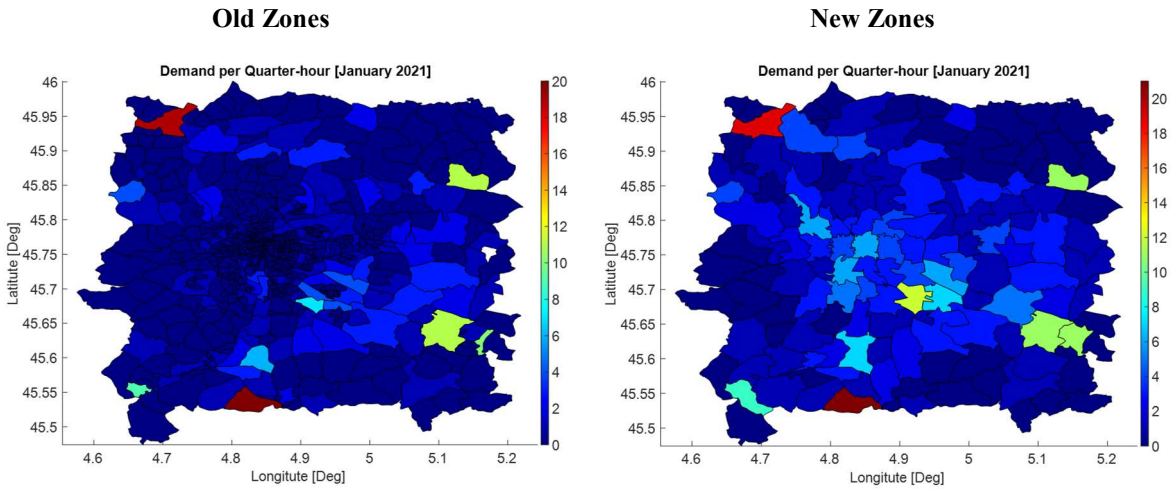


**Fig. 5.** The spatial distribution of the average demand per quarter-hour for the old and new zones in January 2021

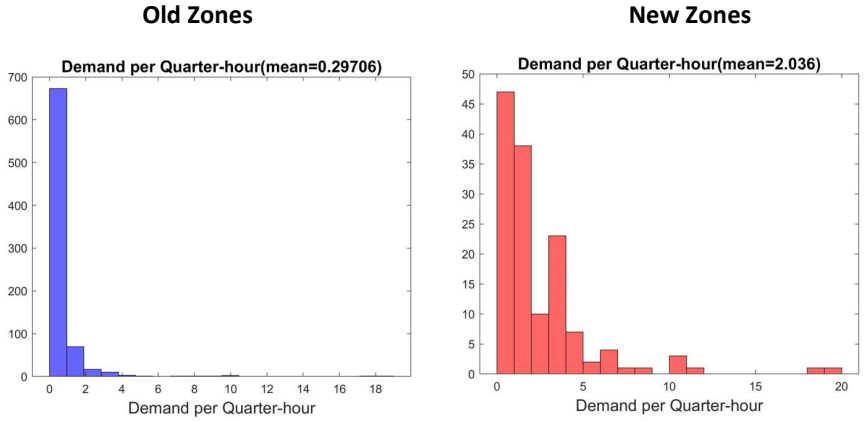


**Fig. 6.** The histograms of the demand per quarter-hour for the old and new zones in January 2021

## 3.3. Experimental setup

The LSTM approach can theoretically be trained with arbitrary sequence lengths. In the current research, the time-step length was assumed to be equal to two values, a quarter-hour and an hour, and the training data length was three weeks. Therefore, considering a quarter-hour and an hour as the time-step lengths, the length of the training sets $N_{\text{Train}}$ (the number of timeslots in the training data sets) were 3×7×24×4 and 3×7×24, respectively (i.e., for a quarter-hour and an hour as the time-step lengths, the time series belonging to each zone includes 2016 and 504 samples on the time axis, respectively). On the other hand, the length of the validation data set (i.e., the number of timeslots in the validation data set) belonging to each zone was considered to include 60 samples (i.e., the data in 15 and 60 hours after the training data were considered

as the validation data sets for the time steps quarter-hour and one-hour, respectively). The rest of the data was assigned to the test dataset which contained 612 and 108 time-steps for quarter-hour and one-hour time-step lengths, respectively (see Fig. 2).

Therefore, based on the prepared data for the Lyon zones, experiments were conducted to predict the traffic demand using the proposed method. In the deep learning stage of the proposed approach, the hyperparameters associated with the LSTM network should be well-designed since adjusting and optimizing these parameters can enhance the performance of the predictions. The most critical hyperparameters of the LSTM network are the number of hidden layers, the number of hidden units per layer, the initial learning rate, and the dropout rate. In the current study, for performing the initial predictions by means of the LSTM network, we used the following values for the hyperparameters (see Table 1). Herein, in order to have the best performance for the initial predictions, these hyperparameters have been tuned via the Bayesian optimization (Snoek et al., 2012) by using the validation part of the data. Bayesian optimization as a powerful strategy, which employs Bayes Theorem to lead the search to the minimum or maximum of an objective function, is usually employed by machine learning algorithms to tune the hyperparameters of a given model (e.g., see (Gelbart et al., 2014; Snoek et al., 2012)).

**Table 1**

The hyperparameters optimization required for the LSTM predictions

| Hidden layers | Hidden units | Initial learning rate | Dropout rate |
|---|---|---|---|
| 1 | 200 | 0.005 | 0.2 |

## 3.4. Performance metrics

In the numerical experiments, we needed certain metrics to evaluate the proposed approach's performance. Such performance metrics have been introduced in the literature in order to measure the quality of the predictions obtained by different approaches. Herein, two widely used metrics, called Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), were used to evaluate the performance of the prediction approaches (Li et al., 2018, 2019; Luo et al., 2021; Yin et al., 2021). For the zone $z_i$, these metrics are defined over the time range as follows:

$$\mathrm{MAPE}(z_i) = \frac{1}{N_{\mathrm{Time}}} \sum_{j=1}^{N_{\mathrm{Time}}} \frac{\left| \hat{d}_{ij} - d_{ij}^{\mathrm{True}} \right|}{\left| d_{ij}^{\mathrm{True}} \right| + c} \tag{16}$$

$$\mathrm{RMSE}(z_i)=\sqrt{\frac{1}{N_{\mathrm{Time}}}\sum_{j=1}^{N_{\mathrm{Time}}}(\hat{d}_{ij}-d_{ij}^{\mathrm{True}})^2} \tag{17}$$

where $N_{\mathrm{Time}}$ is the number of the time slots at which the demand has been predicted, $d_{ij}^{\mathrm{True}}$ is the true value of the demand in the zone $i$ at time-step $t_j=j\Delta t$, and $\hat{d}_{ij}$ is the predicted demand. The Constant $c$ is a small number to avoid division by zero when both $d_{ij}^{\mathrm{True}}$ and $\hat{d}_{ij}$ are zero (herein, it is considered that $c=1$). Similarly, for examining the prediction performance over the entire area of interest, these metrics can be evaluated over all the zones at a time step as follows:

$$\mathrm{MAPE}(t_j)=\frac{1}{N_{\mathrm{Zone}}}\sum_{i=1}^{N_{\mathrm{Zone}}}\frac{\left|\hat{d}_{ij}-d_{ij}^{\mathrm{True}}\right|}{\left|d_{ij}^{\mathrm{True}}\right|+c} \tag{18}$$

$$\mathrm{RMSE}(t_j)=\sqrt{\frac{1}{N_{\mathrm{Zone}}}\sum_{i=1}^{N_{\mathrm{Zone}}}(\hat{d}_{ij}-d_{ij}^{\mathrm{True}})^2} \tag{19}$$

where $N_{\mathrm{Zone}}$ is the number of zones.

In addition to the above-mentioned performance metrics, in the current research, we also defined another performance metric based on the absolute error, and we employed it in our case study. We call this performance metric Percentage of Success (PS) and present it in a tabular form. This metric can give us more details on errors occurring in each forecasted time series. In fact, the PS metric presents us with the percentage of success in predicting each time series for a specific time interval and error threshold. Hence, the PS metric is defined as follows:

**Definition 2 (Percentage of Success (PS)):** For an error threshold $\delta>0$ and a value $\rho\in[0,100]$ called the percentage of accepted time intervals, $\mathrm{PS}(\delta,\rho)$ as the Percentage of Success (PS) metric is defined by the percentage of the zones at which more than $\rho$ percent of the time intervals have been accepted with the absolute error less than $\delta$. In other words, the PS metric is defined as follows:

$$\mathrm{PS}(\delta,\rho)=\frac{N_{\delta,\rho}}{N_{zone}}\times 100 \tag{20}$$

where $N_{zones}$ is the number of zones and $N_{\delta,\rho}$ is the number of zones at which the percentage of accepted time intervals are more than $\rho$. Each time interval and its associated predicted

demand is considered to be accepted if the absolute error of the predicted demand is less than $\delta$. So, the number $N_{\delta,\rho}$ can be counted via the following formula:

$$N_{\delta,\rho}=\sum_{i=1}^{N_{Zones}}H\left(\frac{100}{N_{Time}}\sum_{j=1}^{N_{Time}}H\left(\delta-\left|\epsilon_{ij}\right|\right)-\rho\right) \tag{21}$$

where $\left|\epsilon_{ij}\right|$ is the absolute error of the predicted demand of the $i^{th}$ zone ($z_i$) at the $j^{th}$ time slot ($t_j$) (i.e., $\left|\epsilon_{ij}\right|=\left|\hat{d}_{ij}-d_{ij}^{\text{True}}\right|$), $N_{Time}$ is the number of the time slots at which the demand has been predicted, and $H\left(x\right):\mathbb{R}\rightarrow\left\{0,1\right\}$ is a Heaviside step function (a unit step function) defined as follows:

$$H\left(x\right)=\begin{cases}1 & x\geq 0\\ 0 & x<0\end{cases} \tag{22}$$

## 3.5. Experiments and results

In this section, we compared our approach's final predictions with predictions of the initial LSTM and initial hierarchical reconciliation approaches. At first, this section tries to demonstrate the impact of considering the external features of time (time-of-day and day-of-week) as another input of the LSTM network (in addition to historical demands) on the predictions of travel demand. Then it presents us with the numerical validations of the proposed approach. Fig. 7 illustrates the average RMSE of forecasts in different zones against the time (see Eq.(19)), produced by two LSTM approaches 1) the LSTM network that only uses the historical demand (called Initial LSTM) and 2) the LSTM network that uses both the historical demand and the external features of time (called LSTM Considering Time). The presumed IRIS zones and the aggregated zones were considered as old and new zones, respectively. Furthermore, the results were provided by the two values of the time step, fifteen minutes (see Fig. 7 (a) and (b)) and an hour (see Fig. 7 (c) and (d)).

As seen in Fig. 7, for the time resolution (time-step length) of fifteen minutes, the LSTM considering the external features of time as an additional input could outperform the initial LSTM considering only the historical demand, with the 37.78% and 34.9% improvement in the mean value of the RMSE for the old and new zones, respectively. In addition, for the time resolution (time-step length) equal to one hour, such improvement in the mean value of RMSE was 27.39% and 17.39% for the old and new zones, respectively.

Similarly, Fig. 8 illustrates the comparison in terms of the MAPE performance metric (see Eq.(18)). As shown in Fig. 8, the LSTM network considering the time as a supplementary input could also enhance the MAPE performance metric. We can see that the LSTM approach considering the time as another input could respectively improve the MAPE of the old and new zones by more than 14% and 20% when the time resolution was 15 minutes and could enhance their MAPE performance metrics by more than 11% and 18% when the time resolution was equal to one hour.

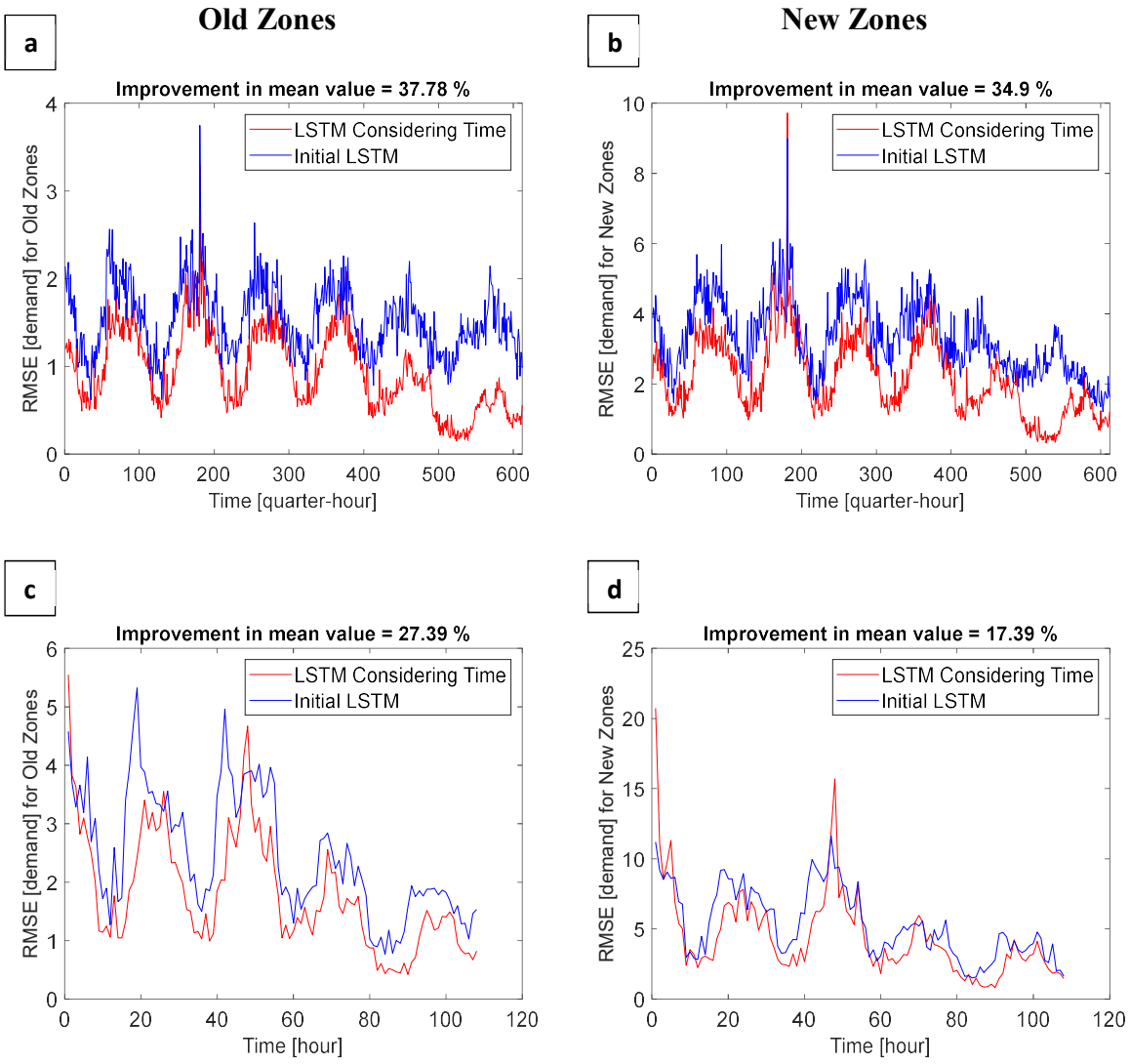


**Fig. 7.** The RMSE performance comparison of two LSTM networks: the comparison between the initial LSTM, which considers only the historical demand, with the LSTM that considers both the historical demand and the external features of time (time-of-day and day-of-week) for the IRIS zones as old zones, and the aggregated zones as new zones, with the time resolutions of 15 minutes (a and b), and an hour (c and d)

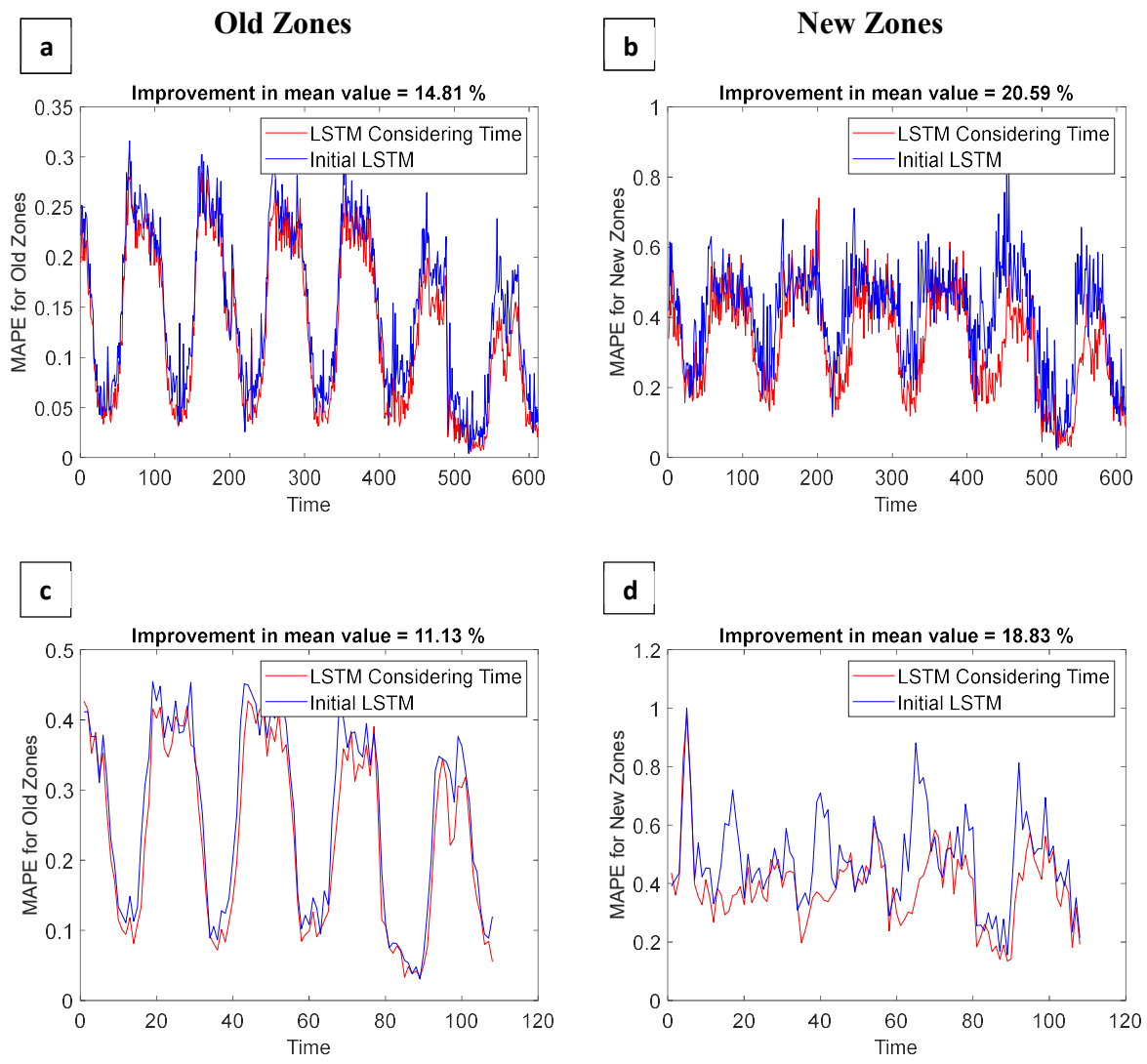


**Fig. 8.** The MAPE performance comparison of two LSTM networks: the comparison between the initial LSTM, which considers only the historical demand, with the LSTM that considers both the historical demand and the external features of time (time-of-day and day-of-week) for the IRIS zones as old zones, and the aggregated zones as new zones, with the time resolutions of 15 minutes (a and b), and an hour (c and d)

Then, to investigate the effectiveness of using both the hierarchical reconciliation and filtering parts of the proposed method, we compared the results of predictions obtained by the proposed method with the initial predictions provided by the LSTM using both the historical demand and the external features of time. As shown in Figures 9 and 10, the proposed method could outperform the initial forecasts of the LSTM that even used both the historical demand and the external features of time, in terms of both RMSE and MAPE metrics for both time resolutions,

fifteen minutes and one hour. The improvement value of the proposed method, which is based on the mean value of the metrics calculated within the prediction interval, is mentioned in each figure, and such enhancement in the results can numerically demonstrate the importance of reconciliation and filtering in the proposed approach.

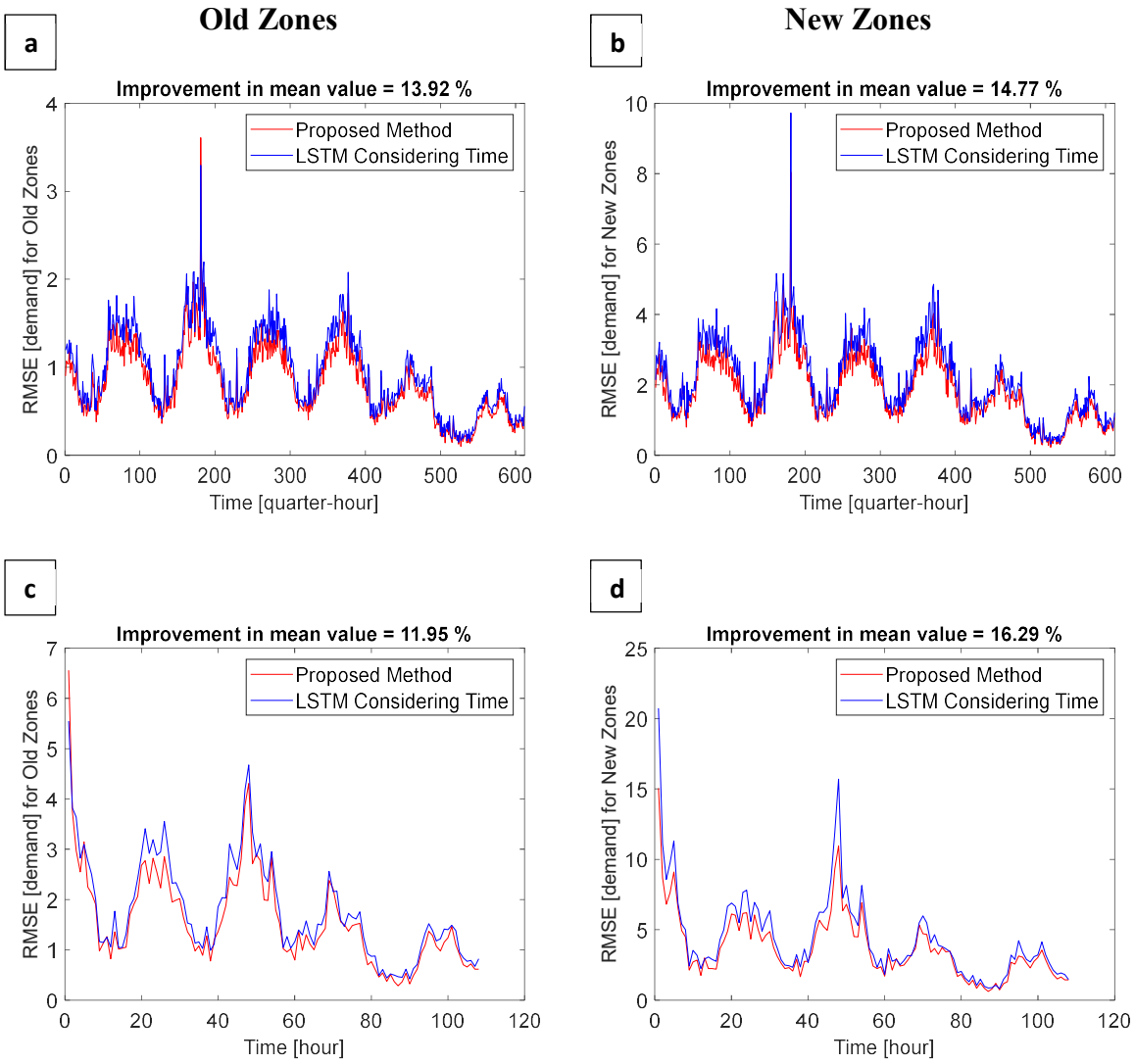


**Fig. 9.** The comparison of the RMSE performance metric of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) for the IRIS zones (old zones) and aggregated zones (new zones) with the time resolution of fifteen minutes (Figures a and b) and an hour (Figures c and d)

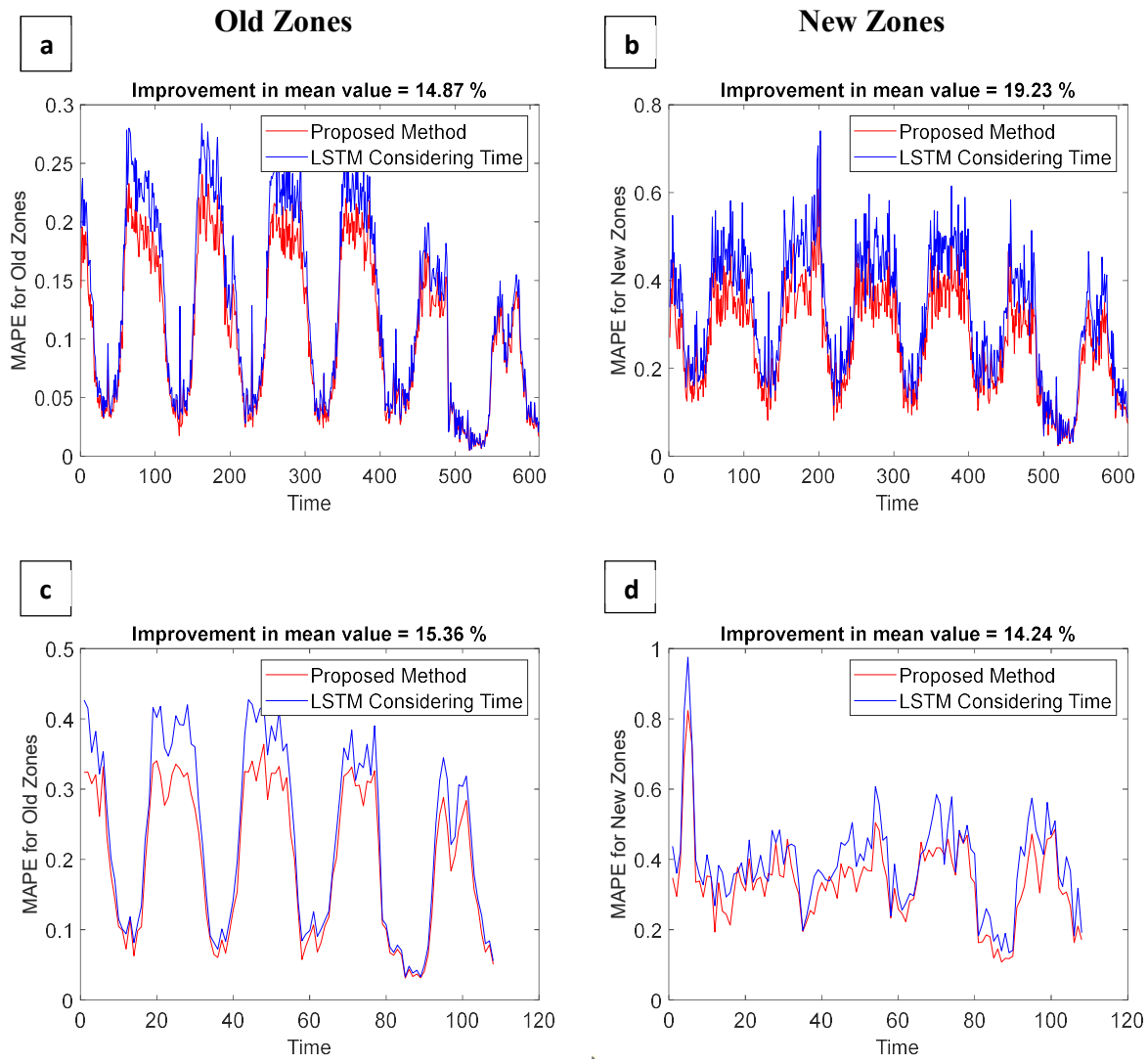


**Fig. 10.** The comparison of the MAPE performance of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) for the IRIS zones (old zones) and aggregated zones (new zones) with the time resolution of fifteen minutes (Figures a and b) and an hour (Figures c and d)

To assess the necessity of the optimal reconciliation, we compared the proposed method with the Bottom-Up approach applied to the demand predictions of old zones obtained by the LSTM that considers the historical demand, as well as the external features of time, to produce the demand predictions of the aggregated zones (new zones). Although the external features of time were also considered in the Bottom-Up approach, the results indicated that the proposed method accompanying the optimal reconciliation could outperform the Bottom-Up approach in predicting the demands of new zones (see Table 2 and Fig. B.1). By comparing the equations

of the Bottom-Up approach with the ones of the optimal reconciliation, we can easily realize that the equations of the optimal reconciliation are the generalized form of the ones of the Bottom-Up approach, which can enable us to seek the optimal solutions in a larger space by means of all the initial demands, while the Bottom-Up approach looks for the solutions of the top level just within the Bottom level without using any information of the other levels.

**Table 2**
The percentage of improvement in the performance metrics' mean value of the proposed approach in comparison with the other approaches

| | **Old Zones** | | | | **New Zones** | | | |
|---|---|---|---|---|---|---|---|---|
| | **15 minutes** | | **One hour** | | **15 minutes** | | **One hour** | |
| | **RMSE** | **MAPE** | **RMSE** | **MAPE** | **RMSE** | **MAPE** | **RMSE** | **MAPE** |
| **Bottom-Up** | - | - | - | - | 12.61 | 17.92 | 10.45 | 13.26 |
| **LSTM+HR_OLS** | 10.94 | 12.58 | 9.65 | 14.44 | 10.05 | 14.83 | 8.92 | 10.76 |
| **LSTM+HR_WLS** | 11.23 | 12.68 | 9.11 | 14.2 | 8.29 | 13.22 | 6.54 | 9.57 |

As we mentioned before, the weight matrix is needed to optimally reconcile the demands in all the zones. If we suppose that the weight matrix is equal to an identity matrix (I), such a case of the problem can be referred to as an Ordinary Least Square (OLS) when there exist no weights between the observations. Applying this approach is straightforward and fast but considers neither the statistical correlations between the time series nor their accurate uncertainties.

Table 2 shows the results of the proposed method in comparison with the results of ordinary least square hierarchical reconciliation (called LSTM+HR_OLS) in terms of the RMSE and MAPE (see also Figures B.2 and B.3). As we can see in this table (as well as the mentioned figures), the proposed method could make more accurate predictions. Regarding the RMSE, the average performance of the forecasts by the time step of fifteen minutes has improved by 10.94% and 10.05% for the old and new zones, respectively. By the time step of one hour, the improvement is equal to 9.65% for the old zones and 8.92% for the new zones (see Table 2 and Fig. B.2).

Regarding the MAPE, the average performance of the demands forecasted via the proposed method for fifteen minutes time resolution compared with OLS hierarchical reconciliation has improved by 12.58% and 14.83% for old and new zones, respectively. In addition, the improvement for one-hour time resolution is 14.44% for old zones and 10.76% for new zones (see Table 2 and Fig. B.3).

Considering the weight matrix as Eq.(11), the approach can be referred to as Weighted Least Square (WLS). To investigate the effectiveness of the filtering part of the proposed approach, we compared the demand forecasts of the proposed method with the ones obtained just by applying the WLS hierarchical reconciliation to the LSTM initial predictions (without applying the filtering stage to them), considering the historical demand and the external features of time (called LSTM+HR_WLS). As in Table 2 (as well as Figures B.4 and B.5), the filtering part of the proposed approach could enhance the results and cause certain improvements in the average RMSE and MAPE of old and new zones.

Finally, based on the Percentage of Success (PS) metric (see Definition 2), we could perform another comparison between the performance of the proposed approach, the Initial LSTM, and the LSTM Considering Time.

**Table 3**
The performance of initial LSTM considering the historical demand in terms of Percentage of Success for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes and one hour (%)

| Percentage of Success (PS) [%] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Old Zones | | | | New Zones | | | |
| | | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands |
| 15 minutes | 90% < AIs | 86 | 95 | 97 | 98 | 38 | 65 | 75 | 86 |
| | 70% < AIs | 95 | 98 | 99 | 99 | 64 | 83 | 96 | 96 |
| | 50% < AIs | 97 | 99 | 99 | 100 | 82 | 96 | 97 | 99 |
| 1 hour | 90% < AIs | 63 | 84 | 92 | 95 | 17 | 38 | 56 | 63 |
| | 70% < AIs | 87 | 95 | 97 | 99 | 41 | 62 | 78 | 90 |
| | 50% < AIs | 95 | 99 | 99 | 100 | 64 | 88 | 95 | 99 |

Tables 3, 4, and 5 present the Percentage of Success (PS) for the proposed method and the two cases of the LSTM. In the first case, just the historical demand has been considered as the input of the LSTM network for forecasting the demands, and in the second case, both the historical demand and the external features of time (time-of-day and day-of-week) have been adopted as the input. As seen in these tables, the Percentage of Success has been computed for four values of the error threshold $\delta$ (2, 4, 6, and 8 demands) and three values of $\rho$, the percentage of accepted time intervals (50%, 70%, and 90%). Moreover, all the PS values have been evaluated for the two values of the time step length (the time resolution), i.e., fifteen minutes and one hour. Accordingly, by comparing the table associated with the proposed method with the ones of the LSTM, we can easily see that for all the values of the error threshold $\delta$ and percentage

of accepted intervals (called AIs) $\rho$, the proposed approach possesses the highest values of the Percentage of Success among the two cases of the LSTM approach. In addition, the comparison of Tables 3 and 4 shows the impact of considering the external features of time. As seen in these tables, regarding the PS metric, the LSTM considering the external features of time as an additional input could outperform the initial LSTM considering only the historical demand.

**Table 4**
The performance of the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) in terms of Percentage of Success for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes and one hour (%)

| Percentage of Success (PS) [%] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Old Zones | | | | New Zones | | | |
| | | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands |
| 15 minutes | 90% < AIs | 89 | 96 | 99 | 99 | 42 | 67 | 91 | 95 |
| | 70% < AIs | 97 | 99 | 100 | 100 | 69 | 94 | 99 | 99 |
| | 50% < AIs | 99 | 100 | 100 | 100 | 93 | 99 | 100 | 100 |
| 1 hour | 90% < AIs | 67 | 88 | 94 | 97 | 20 | 42 | 61 | 67 |
| | 70% < AIs | 90 | 97 | 99 | 100 | 47 | 70 | 91 | 96 |
| | 50% < AIs | 97 | 99 | 100 | 100 | 66 | 96 | 99 | 99 |

**Table 5**
The performance of the proposed method in terms of Percentage of Success for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes and one hour (%)

| Percentage of Success (PS) [%] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Old Zones | | | | New Zones | | | |
| | | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands | Error <= 2 demands | Error <= 4 demands | Error <= 6 demands | Error <= 8 demands |
| 15 minutes | 90% < AIs | 92 | 98 | 99 | 99 | 51 | 83 | 93 | 96 |
| | 70% < AIs | 98 | 99 | 100 | 100 | 80 | 96 | 99 | 100 |
| | 50% < AIs | 99 | 100 | 100 | 100 | 94 | 99 | 100 | 100 |
| 1 hour | 90% < AIs | 72 | 91 | 96 | 98 | 22 | 49 | 64 | 78 |
| | 70% < AIs | 91 | 98 | 99 | 100 | 52 | 80 | 95 | 98 |
| | 50% < AIs | 98 | 100 | 100 | 100 | 75 | 97 | 99 | 100 |

# 4. Conclusions

In this research, an efficient method for traffic demand forecasting was developed via deep learning and hierarchical reconciliation approaches. Herein, we founded the required concepts and theories on Hierarchical Time Series (HTS), where we also employed Long Short-term Memory (LSTM) for the deep learning and initial forecasting of the associated time series. The proper design of the HTS structure enables the proposed approach to present coherent forecasts for the demand over predefined zones and aggregated zones for the purposes of mobility service operations. Also, deep learning enables the HTS structure to provide reliable predictions of the demands. Therefore, the approach propounded by the current research is based on three main stages: 1) deep learning, 2) error analysis, and 3) optimal reconciliation, where the error analysis step makes a connection between the first and last steps (deep learning and optimal reconciliation).

Based on the presented theories and concepts (see Section 2), the initial forecasts obtained by the LSTM are not ready to be directly used by the optimal reconciliation step. Thus, the error analysis prepares the initial predictions to be converted to the final forecasts via optimal reconciliation. The error analysis controls the remaining errors of the initial prediction attained via the LSTM deep learning and prevents their time series from diverging via a filtering method. Also, the error analysis finally provides the weight matrix essentially required by the optimal reconciliation. Such abilities of the error analysis stage can also enable the proposed method to employ other deep learning methods for initially forecasting the demand, where the error analysis step can analyze, monitor, and prepare the initial forecasts for the hierarchical reconciliation.

Meanwhile, in this research, concerning the requirements of the demand prediction, we needed to aggregate the zones before performing any numerical experiments. Hence, we also proposed a method for aggregating the zones in an area based on the least square method. Herein, we had a predefined set of IRIS zones, and based on the spatial information of such IRIS zones and temporal thresholds, we carried out the aggregation to obtain a new set of zones with an average travel time between 5 to 10 minutes in each zone. The procedures and concepts of this approach are presented in Section 3 in detail. It is worth mentioning that our proposed approach is flexible concerning spatial zoning.

Furthermore, to examine the proposed approach's abilities and performance, we also conducted various numerical experiments and comparisons. Thus, since the proposed approach was

founded on its above-mentioned main stages, we tried to assess the proposed method at each stage and numerically examine the necessity of each step. For this purpose, the LSTM approach was considered as the base approach for initially forecasting the demands. Then we investigated how each step of the proposed approach can enhance the initial predictions of the LSTM. In the numerical experiments, different performance metrics have been used to compare the approaches and assess the proposed method's efficiency. Also, this research introduced a performance metric called Percentage of Success (PS) for evaluating the abilities of the methods in predicting the demands. The proposed method could surpass different cases of the LSTM approach and present the best performance metrics in all the comparisons. For instance, according to the defined performance metric PS, the percentage of success for the proposed approach could gain the highest values in all the cases of the numerical experiments compared with the LSTM approach.

# Appendix A. Details on the aggregation of zones

For clustering and aggregating the IRIS zones into ones with the same travel time (here, 5 to 10 minutes in each zone), we needed to analyze and look at travel speed data for the area of interest. Fig. A.1 shows the distribution of the speed in the city of Lyon as the desired area, extracted from the road network of the floating car dataset introduced in Section 3.1. This speed is the average speed in a non-congested situation, in other words, the free-flow speed. According to this data, the average speed in the desired area was around 50 km/h, and the travel distance in 5 minutes was averagely around 4 km. Moreover, this figure illustrates the smoothed values of the speed over the region, where the speed was smoothed by a moving average with a window size equal to 4 km.

In this step, we aimed to precisely convert the regular mesh into a deformed one to have the same travel time for each mesh element. For this purpose, first, we produced a dense uniform grid (a $0.5km \times 0.5km$ regular grid) over the area, and then we sought to find the travel time between all the gridded points. If we could find the travel time between these dense gridded points, we could form the time domain at which the distance between each pair of points represents the travel time between them.

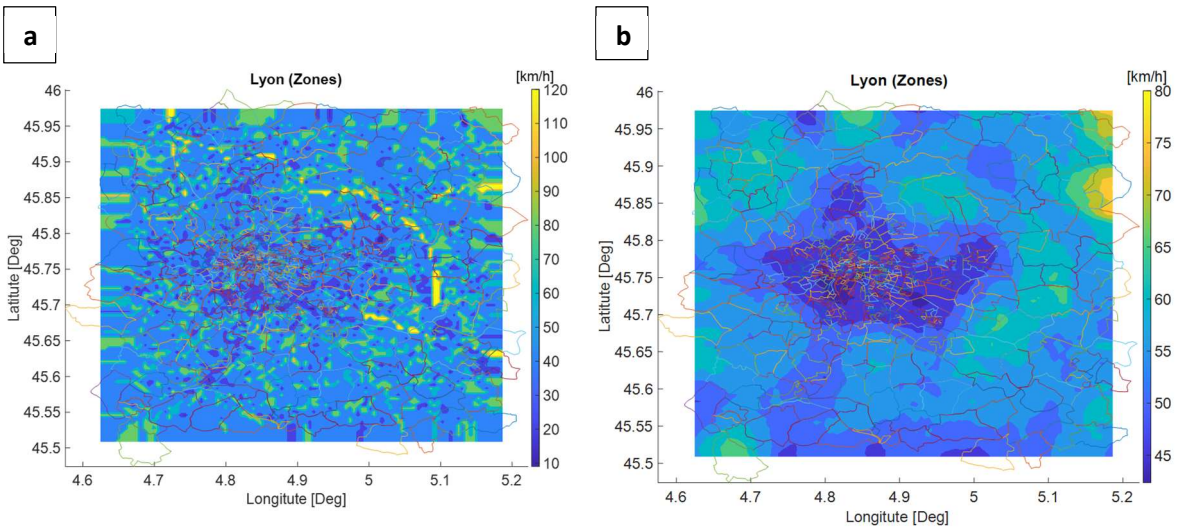


**Fig. A.1.** a) the raw data of the speed over the interest area, extracted from the road network of the floating car dataset, and b) the speed distribution smoothed by a moving average with a 4km window over the Lyon region

In addition, Fig. A.2 shows a $4km \times 4km$ regular mesh over the IRIS zones, demonstrating how the travel time at each square (element) of the mesh depends on the speed distribution.

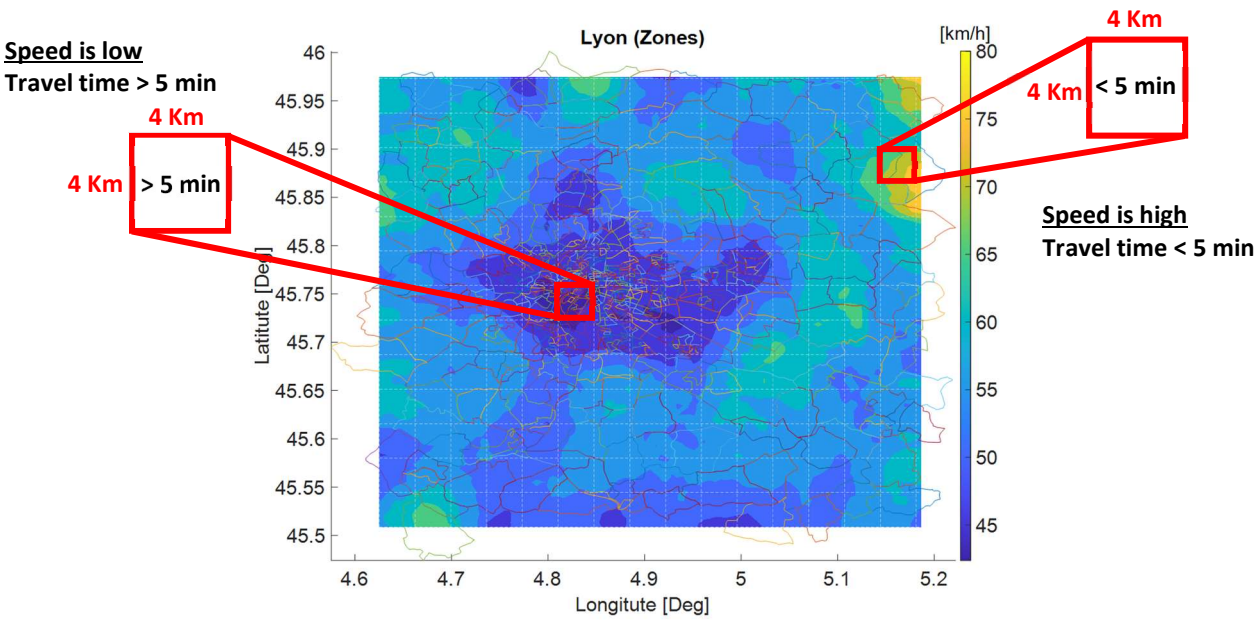


**Fig. A.2.** a) the spatial distribution of speed over the region and a $4km \times 4km$ regular mesh over the IRIS zones: we can see in this figure that the travel time at each square (element) of the mesh depends on the speed distribution

**Definition 3 (Time domain):** herein, a 2D Euclidean space such as $\{(x', y') \in \mathbb{R}^2\}$ is called a time domain where the coordinate difference between each pair of points $(\Delta x', \Delta y')$ indicates the travel time between those two points, and at which the coordinate differences $\Delta x'$ and $\Delta y'$ indicate the travel time in the directions of spatial domain $x$ and $y$, respectively.

Thus, in the spatial domain where the dense grid points form a regular uniform grid, we have $x_{ij} - x_{(i-1)j} = d$ and $y_{ij} - y_{(i-1)j} = d$ where $d = 0.5km$, and $\{(x_{ij}, y_{ij})\}$ are the known locations of the dense grid points in the spatial domain.

On the other hand, we have the following equations between the unknown locations of the points in the time domain, where the distance between each pair of neighboring points indicates the travel time between them. Herein, the velocity information obtained from the road network of the floating car dataset in the Lyon region was used for forming the following equations between each point and its neighboring points. In the following equation, we see that the travel time $\Delta t_{ij}$ between two neighbouring points with indices $ij$ and $(i-1)j$ can be estimated by the known value of the speed $v_{ij}$ at the point $(x_{ij}, y_{ij})$, where $\Delta t_{ij} = d / v_{ij}$. Therefore, we have:

$$\begin{cases} x'_{ij} - x'_{(i-1)j} = \Delta t_{ij} = d / v_{ij} \\ y'_{ij} - y'_{(i-1)j} = \Delta t_{ij} = d / v_{ij} \end{cases} \tag{23}$$

where $\{(x'_{ij}, y'_{ij})\}$ will be the new locations of the dense grid points in the time domain. For finding the new locations of the points in the time domain, we formed the above equation (i.e., Eq.(23)) between each point of the grid and its neighbors, and consequently, a big linear equation could be formed for all the points of the dense grid. Then we could solve this linear equation and find the unknown coordinates $\{(x'_{ij}, y'_{ij})\}$ via the minimum norm solutions of the linear equation obtained by Eq.(23).

Fig. A.3 (a) illustrates the dense grid points in the spatial domain, and similarly, Fig. A.3 (b) illustrates these points in the time domain where the distance between the grid points is equal to the travel time between them, and their new locations in this time domain were obtained via Eq.(23). In this figure, we can easily see that the uniform grid in the spatial domain (Fig. A.3 (a)) experienced a deformation in the time domain (Fig. A.3 (b)), which means that the speed distribution has influenced the locations in the time domain. Besides, we can consider the new locations of these dense grid points as reference points that can assign the new coordinate $(x', y')$ of the time domain to each point of the spatial domain $(x, y)$. Based on such new

coordinates of the reference points in the time domain, we could also achieve the new locations of the zones in the time domain. For this purpose, we just considered the zones' middle points as the zones' representatives. Then the following equation could present us with the new locations of the zones in the time domain, where the proximity and remoteness of the zones are identified by the travel time between them.

$$\begin{cases} x'_{z_n} = x'_{ij} + (x_{z_n} - x_{ij}) / v_{ij} \\ y'_{z_n} = y'_{ij} + (y_{z_n} - y_{ij}) / v_{ij} \end{cases} \quad (24)$$

where $(x_{z_n}, y_{z_n})$ and $(x'_{z_n}, y'_{z_n})$ are the coordinates of the middle point of the zone $z_n$ in the spatial and time domain, respectively. Similarly, $(x_{ij}, y_{ij})$ and $(x'_{ij}, y'_{ij})$ are respectively the known and estimated coordinates of a grid point situated in the vicinity of the middle point of the zone $z_n$. Also, $v_{ij}$ is the travel speed at the point $(x_{ij}, y_{ij})$. Accordingly, Fig. A.4 shows the new locations of the middle points of the IRIS zones in the time domain. In this figure, we can easily identify which zones are closer to each other in terms of travel time.

Then in the time domain, we produced another mesh at which the size of elements (square) was $5\,\text{min} \times 5\,\text{min}$, and based on such a mesh in the time domain, we could perform the clustering and aggregation of the zones. All zones, whose middle points were located in the same $5\,\text{min} \times 5\,\text{min} - \text{element}$, are aggregated with each other and constitute a new zone (see Fig. A.4).

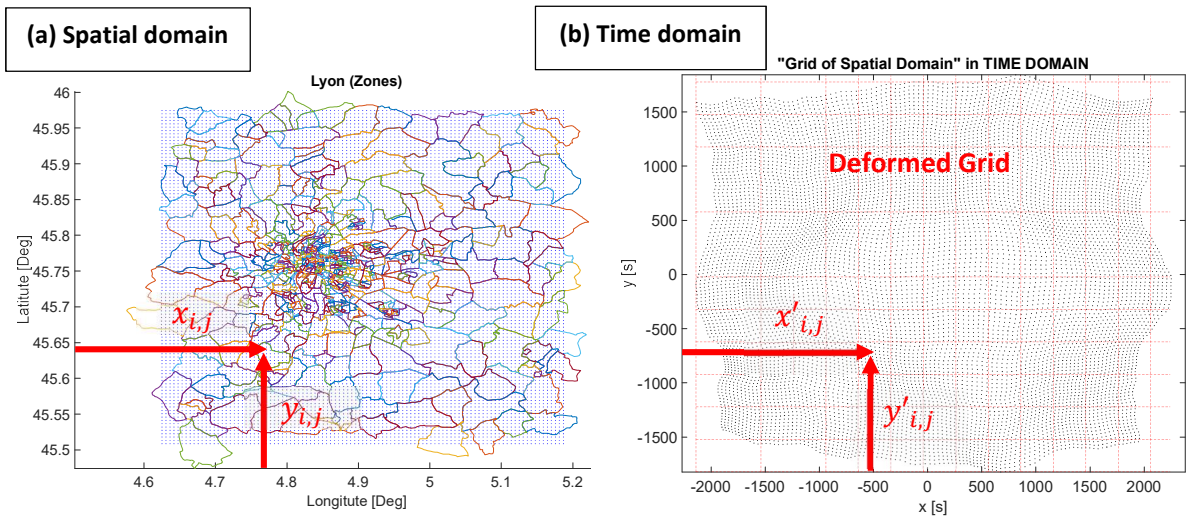


**Fig. A.3.** a) the dense grid points in the spatial domain, defined over the IRIS zones in the Lyon region: these points were defined for relating the spatial domain to the time domain, and b) the time domain where the dense grid points were deformed to obey the travel times and the proximity and remoteness of the points are because of the travel time between them

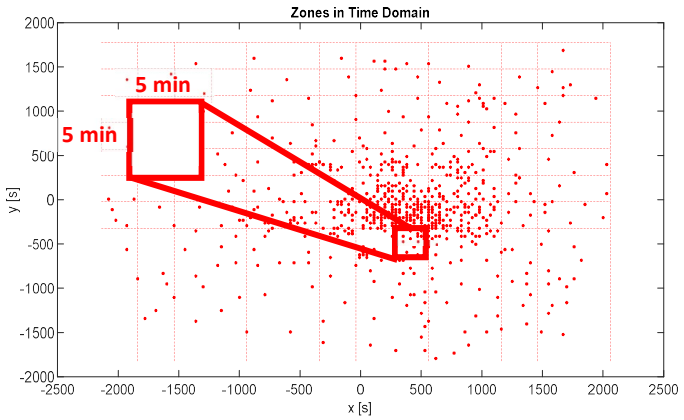


**Fig. A.4.** the zones in the time domain: the red grid lines constitute a 5min×5min mesh in the time domain, where zones situated in each 5min×5min element (square) form an aggregated zone [unit: s]

Moreover, using the reference points (i.e., via the known locations of the dense grid points in both the spatial and time domains), we were also able to transfer the uniform $5\,\mathrm{min}\times 5\,\mathrm{min}$ mesh grid of the time domain to the spatial domain. In Fig. A.5, the deformed mesh grid shows the transferred $5\,\mathrm{min}\times 5\,\mathrm{min}$ mesh grid of the time domain. In this figure, we can clearly see how the $5\,\mathrm{min}\times 5\,\mathrm{min}$ mesh grid coincides with the spatial distribution of the speed. We also see how the $5\,\mathrm{min}\times 5\,\mathrm{min}$-mesh elements shrink in low-speed regions and, on the other hand, how they expand in high-speed areas.

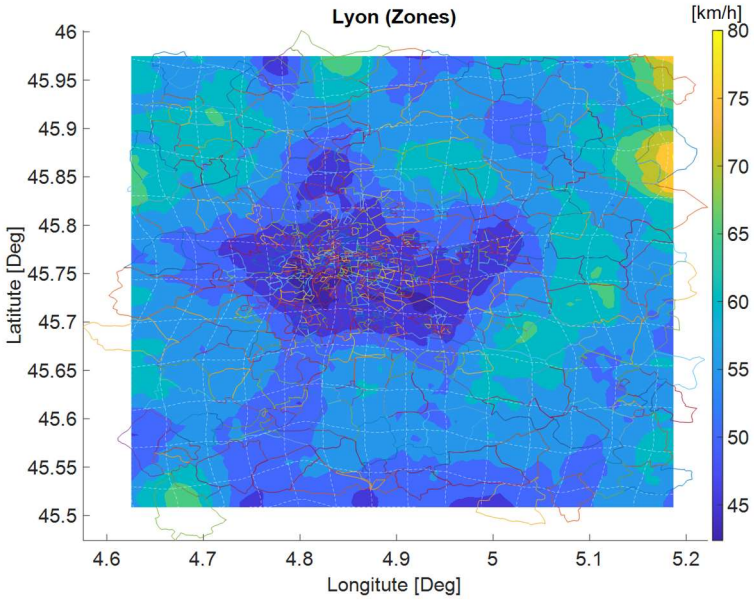


**Fig. A.5.** the deformed 5min×5min-mesh over the Lyon region (the travel time in a deformed element (deformed square) is 5min in each direction): we can see the element size conforms with the speed distribution where the elements shrink and expand in low-speed and high-speed areas, respectively

Accordingly, Fig. A.6 presents us with the new aggregated zones. In this figure, each new aggregated zone is obtained by merging the old zones (the predefined IRIS zones), whose middle points are situated in the same $5\,\mathrm{min}\times 5\,\mathrm{min}$-mesh element. Furthermore, Fig. A.7

illustrates the distribution of the average speed and travel time in the new zones, which indicates that the travel time within the new zones is around 6 minutes, and it is averagely between 2 and 10 minutes for all the new zones.

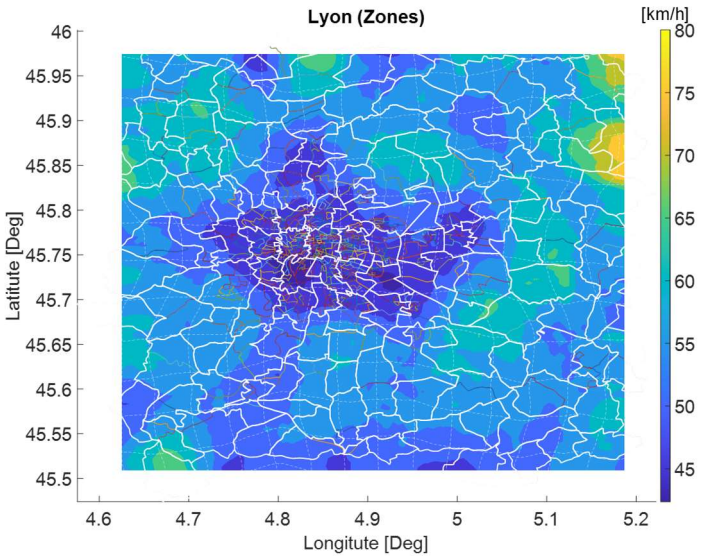


**Fig. A.6.** The new zones obtained by the aggregation of the IRIS zones (old zones)

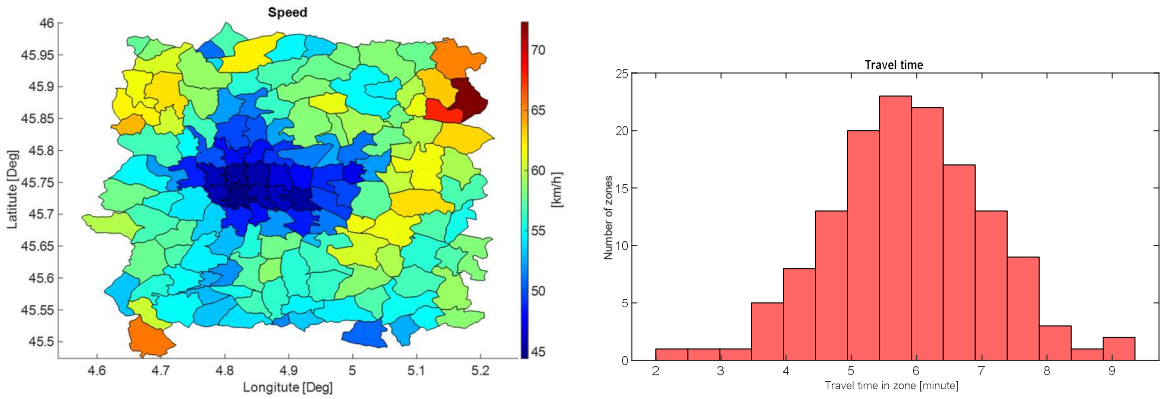


**Fig. A.7.** The average speed and travel time in the new zones

# Appendix B. Plots

Herein, Fig. B.1 illustrates the comparison of the performance of the proposed method and the Bottom-Up approach considering the historical demand and the external features of time (in terms of RMSE and MAPE performance metrics). The LSTM considering the historical demand and the external features of time in combination with ordinary least square hierarchical reconciliation (LSTM+HR_OLS) and the proposed method are compared in Figures B.2 and

B.3. Finally, we can see the comparison of the LSTM considering the historical demand and the external features of time in combination with weighted least square hierarchical reconciliation (LSTM+HR_WLS) and the proposed method in Figures B.4 and B.5. The results are shown for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes and one hour.

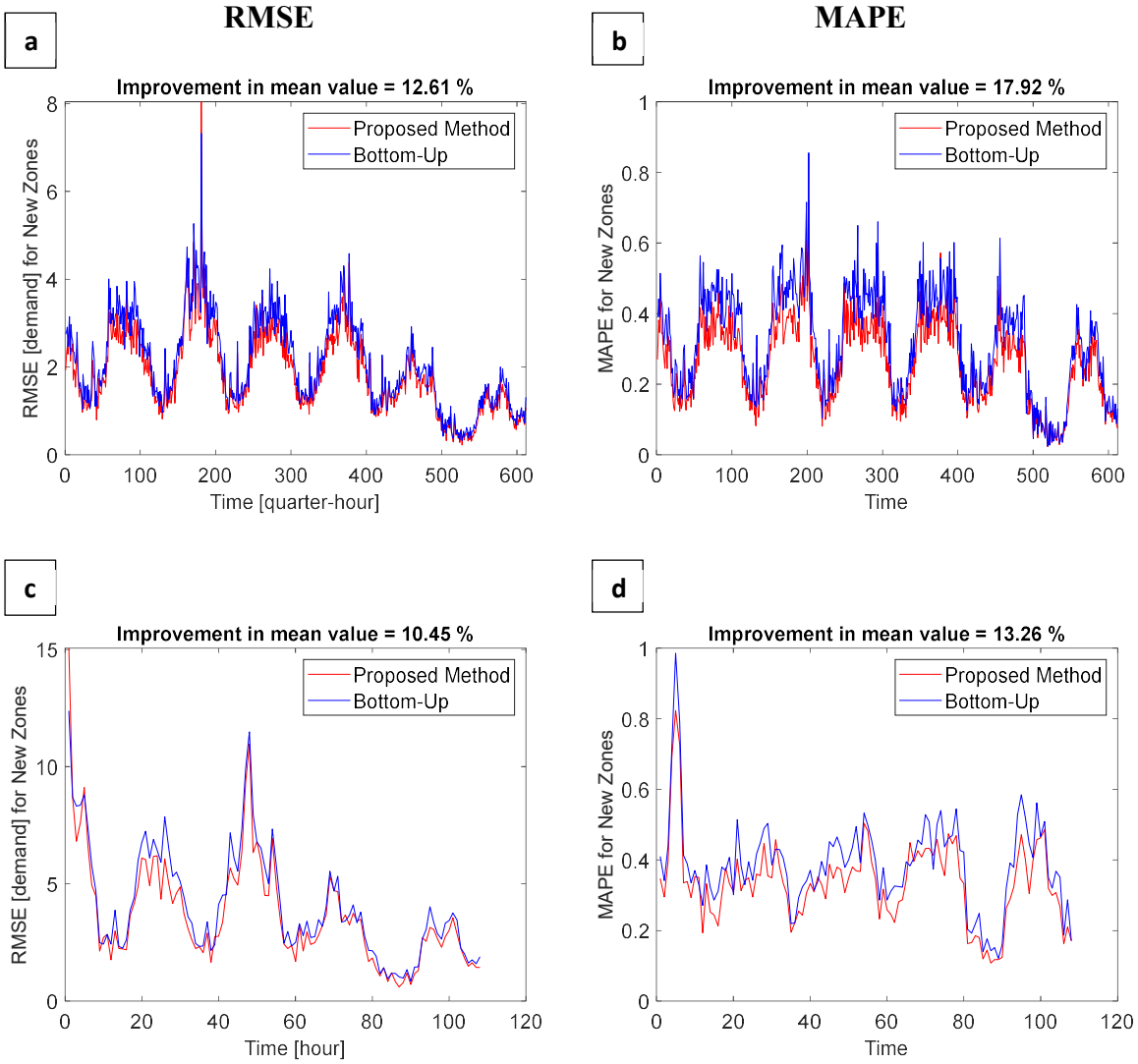


**Fig. B.1.** The comparison of the RMSE performance of the proposed method and Bottom-Up approach considering the historical demand and the external features of time (time-of-day and day-of-week) for the aggregated zones (new zones) with time resolutions of fifteen minutes (Figures a and b) and an hour (Figures c and d)

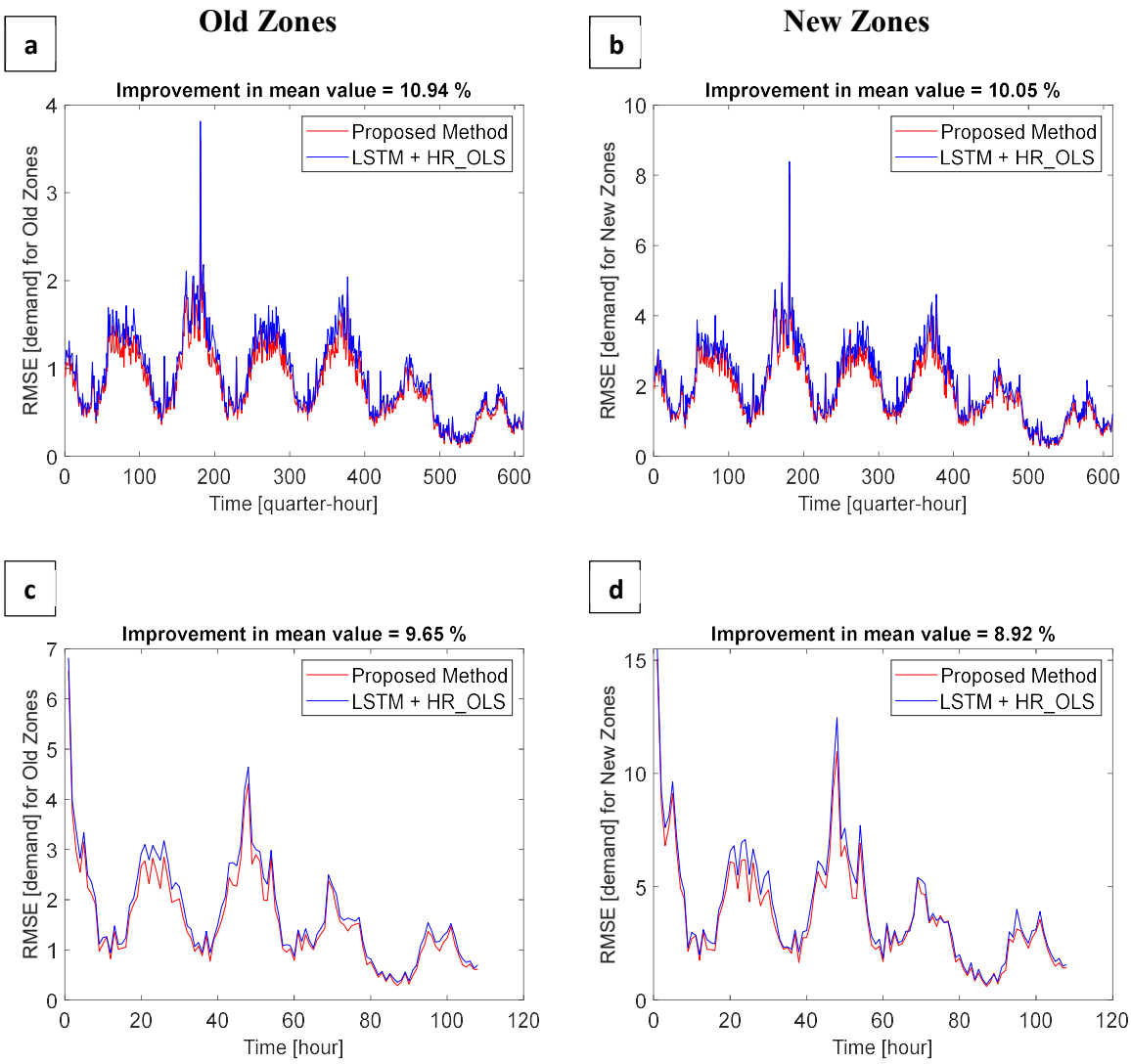


**Fig. B.2.** The comparison of the performance of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) in combination with ordinary least square hierarchical reconciliation (LSTM+HR_OLS) in terms of RMSE performance metric for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes (Figures a and b) and an hour (Figures c and d)

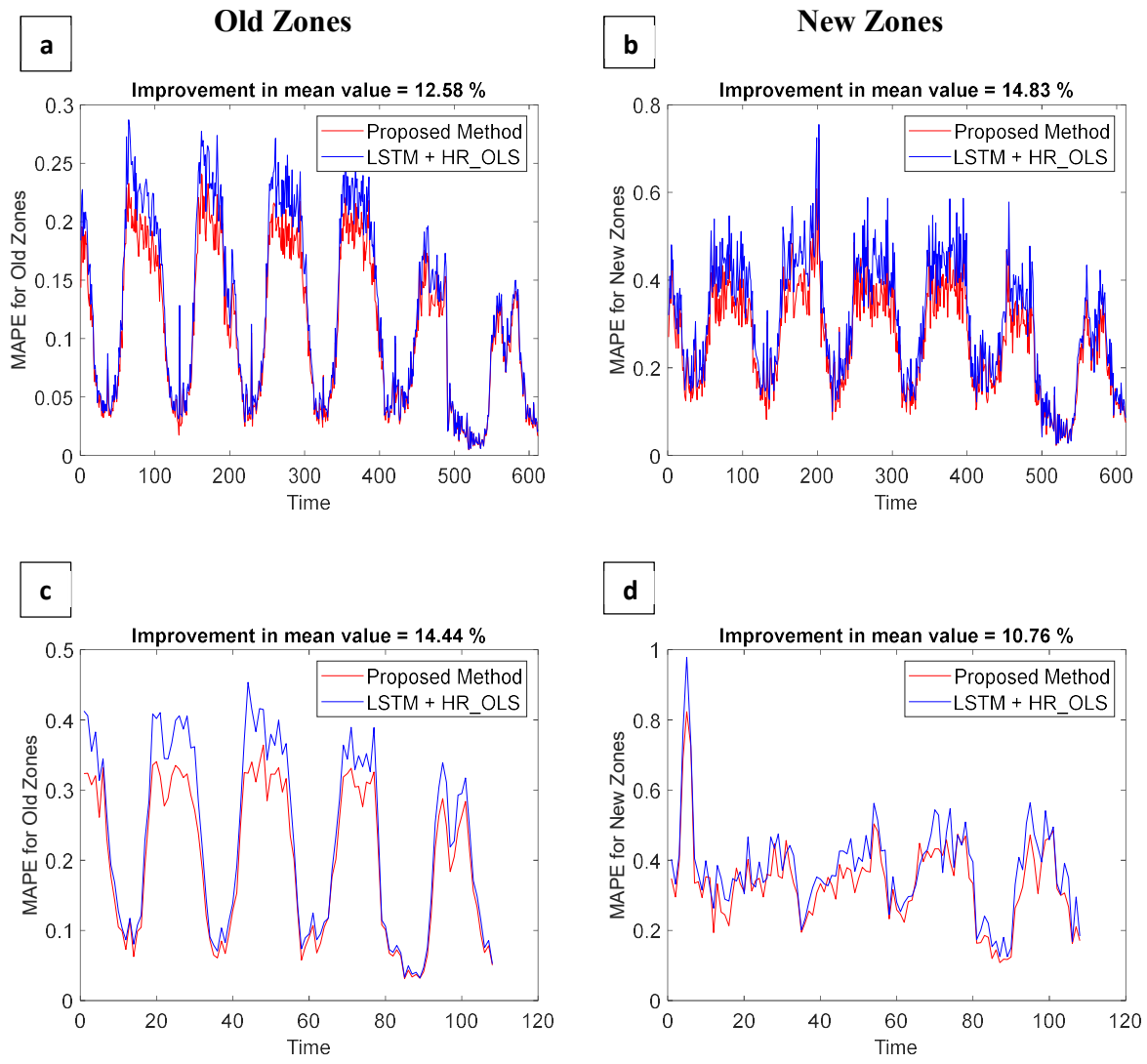


**Fig. B.3.** The comparison of the performance of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) in combination with ordinary least square hierarchical reconciliation (HR_OLS) in terms of MAPE performance metric for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes (Figures a and b) and an hour (Figures c and d)

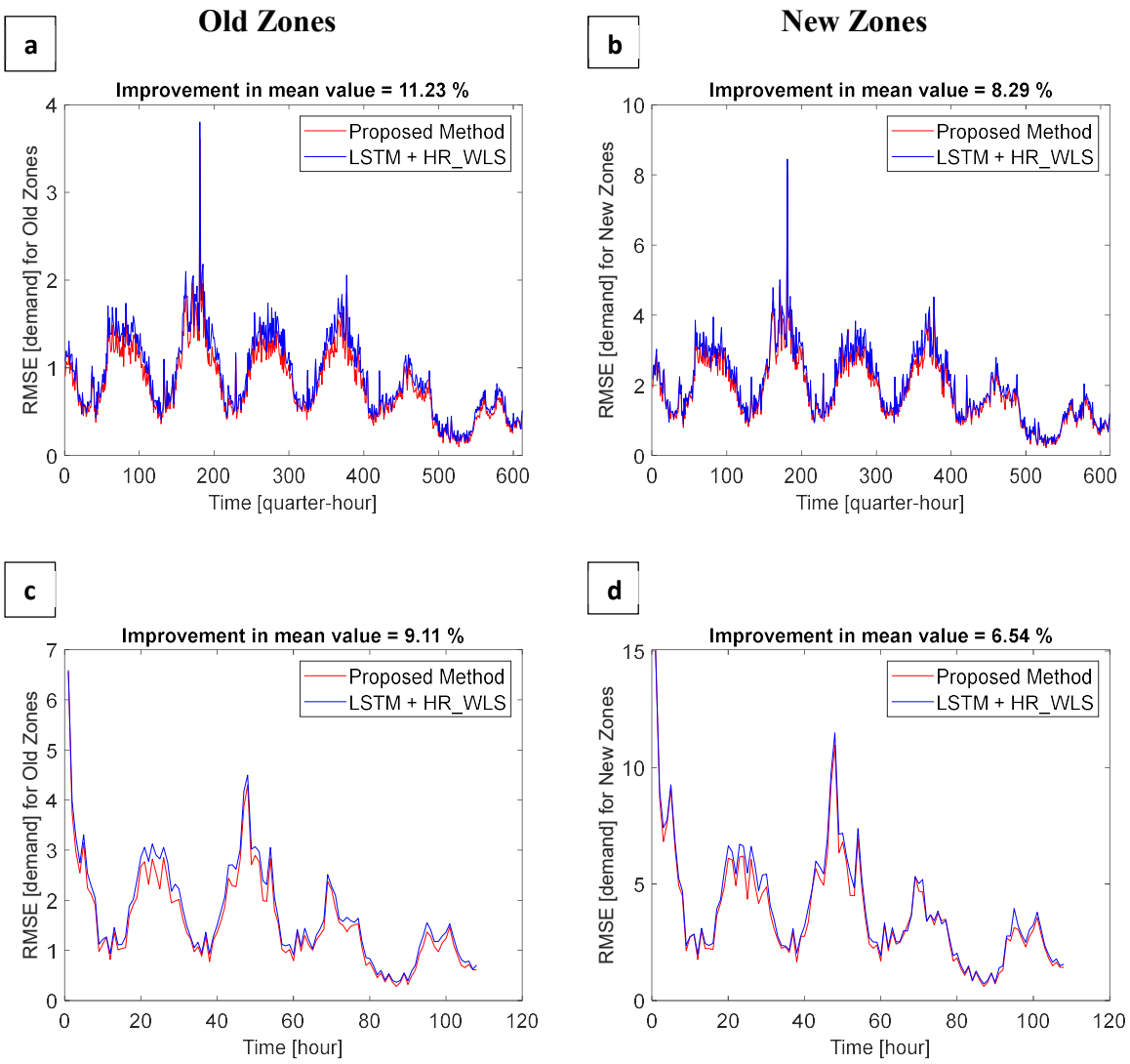


**Fig. B.4.** The comparison of the performance of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) in combination with weighted least square hierarchical reconciliation (HR_WLS) in terms of RMSE performance metric for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes (Figures a and b) and an hour (Figures c and d)

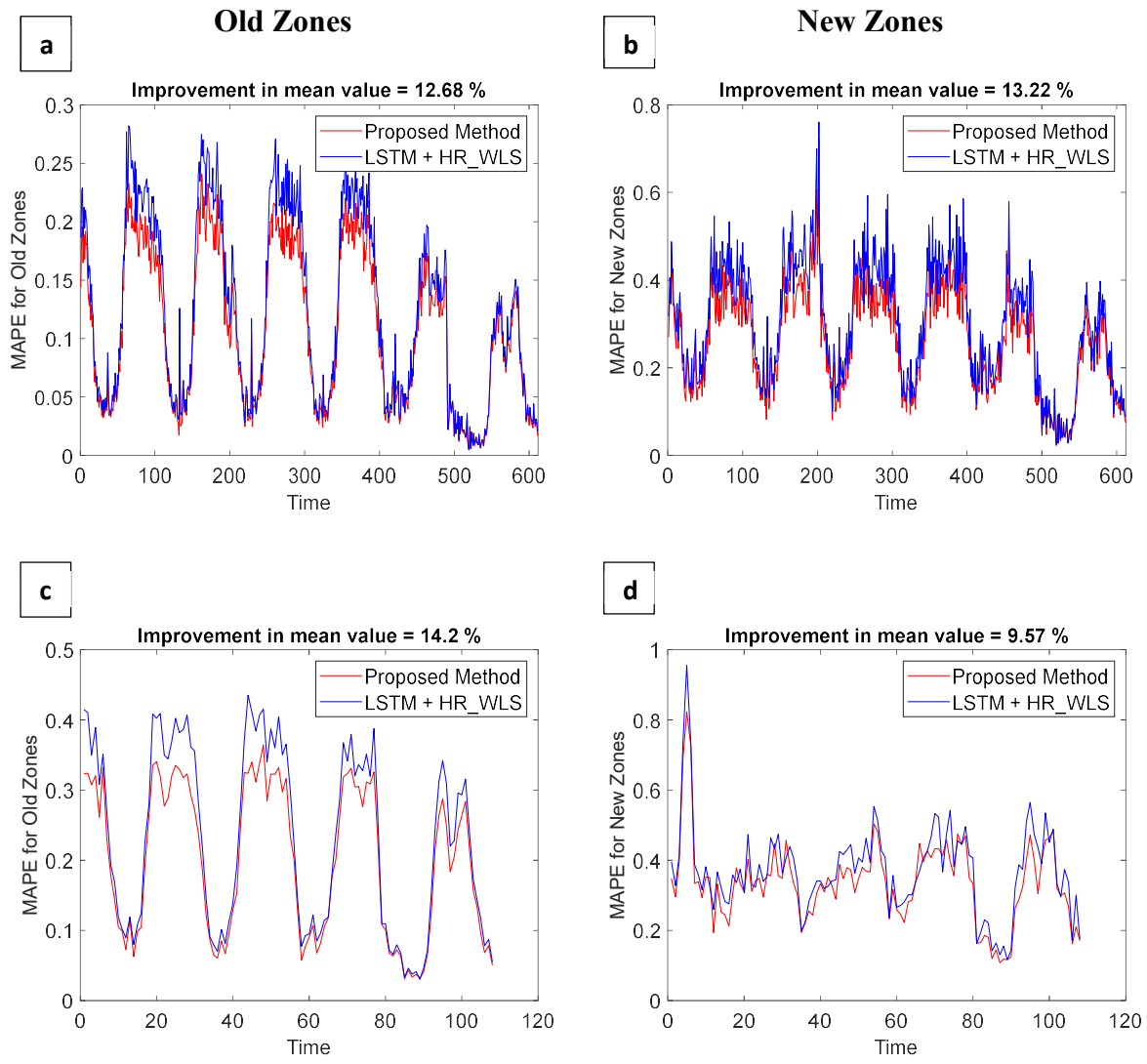


**Fig. B.5.** The comparison of the performance of the proposed method and the LSTM considering the historical demand and the external features of time (time-of-day and day-of-week) in combination with weighted least square hierarchical reconciliation (HR_WLS) in terms of MAPE performance metric for IRIS zones (old zones) and aggregated zones (new zones) with time resolutions of fifteen minutes (Figures a and b) and an hour (Figures c and d)

# References


Athanasopoulos, G., Ahmed, R.A., Hyndman, R.J., 2009. Hierarchical forecasts for Australian domestic tourism. Int. J. Forecast. 25, 146–166.

Banerjee, S., Roy, A., 2014. Linear algebra and matrix analysis for statistics. Crc Press Boca Raton, FL, USA:

Björck, Å., 1996. Numerical methods for least squares problems. SIAM.

Cheng, Q., Liu, Y., Wei, W., Liu, Z., 2016. Analysis and forecasting of the day-to-day travel demand variations for large-scale transportation networks: a deep learning approach. Transp. Anal. Contest Tech Rep.

Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder–decoder approaches. Presented at the Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111.

Connor, J.T., Martin, R.D., Atlas, L.E., 1994. Recurrent neural networks and robust time series prediction. IEEE Trans. Neural Netw. 5, 240–254.

Davis, N., Raina, G., Jagannathan, K., 2020. Grids versus graphs: Partitioning space for improved taxi demand-supply forecasts. IEEE Trans. Intell. Transp. Syst. 22, 6526–6535.

Gelbart, M.A., Snoek, J., Adams, R.P., 2014. Bayesian optimization with unknown constraints. ArXiv Prepr. ArXiv14035607.

Guan, J., Wang, W., Li, W., Zhou, S., 2018. A unified framework for predicting kpis of on-demand transport services. IEEE Access 6, 32005–32014.

Guo, J., Williams, B.M., Smith, B.L., 2007. Data collection time intervals for stochastic short-term traffic flow forecasting. Transp. Res. Rec. 2024, 18–26.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hollyman, R., Petropoulos, F., Tipping, M.E., 2021. Understanding forecast reconciliation. Eur. J. Oper. Res. 294, 149–160.

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L., 2011. Optimal combination forecasts for hierarchical time series. Comput. Stat. Data Anal. 55, 2579–2589.

Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: principles and practice. OTexts.

INSEE & IGN, 2021. IRIS zones. URL https://geoservices.ign.fr/contoursiris

Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transp. Res. Part C Emerg. Technol. 85, 591–608.

Li, A., Axhausen, K.W., 2019. Comparison of short-term traffic demand prediction methods for transport services. Arbeitsberichte Verk.- Raumplan. 1447.

Li, W., Cao, J., Guan, J., Zhou, S., Liang, G., So, W.K., Szczecinski, M., 2018. A general framework for unmet demand prediction in on-demand transport services. IEEE Trans. Intell. Transp. Syst. 20, 2820–2830.

Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., Zhang, W., Wang, Z., 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. Front. Comput. Sci. 6, 111–121.

Li, Y., Lu, J., Zhang, L., Zhao, Y., 2017. Taxi booking mobile app order demand prediction based on short-term traffic forecasting. Transp. Res. Rec. 2634, 57–68.

Li, Z., Zheng, Z., Washington, S., 2019. Short-term traffic flow forecasting: a component-wise gradient boosting approach with hierarchical reconciliation. IEEE Trans. Intell. Transp. Syst. 21, 5060–5072.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., Lin, L., 2019. Contextualized spatial–temporal network for taxi origin-destination demand prediction. IEEE Trans. Intell. Transp. Syst. 20, 3875–3887.

Luo, H., Cai, J., Zhang, K., Xie, R., Zheng, L., 2021. A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependences. J. Traffic Transp. Eng. Engl. Ed. 8, 83–94.

Mancuso, P., Piccialli, V., Sudoso, A.M., 2021. A machine learning approach for forecasting hierarchical time series. Expert Syst. Appl. 182, 115102.

Mikhail, E.M., 1982. Observations and least squares. University Press of Amer.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L., 2013. Predicting taxi–passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. 14, 1393–1402.

Nguyen, H., Kieu, L.-M., Wen, T., Cai, C., 2018. Deep learning methods in transportation domain: a review. IET Intell. Transp. Syst. 12, 998–1004.

Oh, C., Ritchie, S.G., Oh, J.-S., 2005. Exploring the relationship between data aggregation and predictability to provide better predictive traffic information. Transp. Res. Rec. 1935, 28–36.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. nature 323, 533–536.

Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R., Gasthaus, J., 2019. High-dimensional multivariate forecasting with low-rank gaussian copula processes. Adv. Neural Inf. Process. Syst. 32.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2008. The graph neural network model. IEEE Trans. Neural Netw. 20, 61–80.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. Adv. Neural Inf. Process. Syst. 25.

Vlahogianni, E., Karlaftis, M., 2011. Temporal aggregation in traffic data: implications for statistical characteristics and model choice. Transp. Lett. 3, 37–49.

Wickramasuriya, S.L., Athanasopoulos, G., Hyndman, R.J., 2019. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. J. Am. Stat. Assoc. 114, 804–819.

Xu, J., Rahmatizadeh, R., Bölöni, L., Turgut, D., 2017. Real-time prediction of taxi demand using recurrent neural networks. IEEE Trans. Intell. Transp. Syst. 19, 2572–2581.

Yang, H., Leung, C.W., Wong, S.C., Bell, M.G., 2010. Equilibria of bilateral taxi–customer searching and meeting on networks. Transp. Res. Part B Methodol. 44, 1067–1083.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B., 2021. Deep learning on traffic prediction: Methods, analysis and future directions. IEEE Trans. Intell. Transp. Syst.