LSTM encoder-predictor for short-term train load forecasting

Kevin Pasini ✉[1,2], Mostepha Khouadjia[1], Allou Samé[2], Fabrice Ganansia[3], and Latifa Oukhellou[2]

[1] Université Paris-Est, IFSTTAR, Cosys-Grettia, Champs-sur-Marne, France
[2] IRT SystemX, Saclay, France
[3] SNCF- Innovation & Recherche, St Denis, France
Kevin.pasini@irt-systemx.fr

Abstract. The increase in the amount of data collected in the transport domain can greatly benefit mobility studies and help to create high value-added mobility services for passengers as well as regulation tools for operators. The research detailed in this paper is related to the development of an advanced machine learning approach with the aim of forecasting the passenger load of trains in public transport. Predicting the crowding level on public transport can indeed be useful for enriching the information available to passengers to enable them to better plan their daily trips. Moreover, operators will increasingly need to assess and predict network passenger load to improve train regulation processes and service quality levels. The main issues to address in this forecasting task are the variability in the train load series induced by the train schedule and the influence of several contextual factors, such as calendar information. We propose a neural network LSTM encoder-predictor combined with a contextual representation learning to address this problem. Experiments are conducted on a real dataset provided by the French railway company SNCF and collected over a period of one and a half years. The prediction performance provided by the proposed model are compared to those given by historical models and by traditional machine learning models. The obtained results have demonstrated the potential of the proposed LSTM encoder-predictor to address both one-step-ahead and multi-step forecasting and to outperform other models by maintaining robustness in the quality of the forecasts throughout the time horizon.

Keywords: Machine learning · Time series forecasting · Representation learning · Mobility data · Transport · Train load.

1 Introduction

In recent years, the population growth in metropolitan areas has led to overcrowding on trains. Transport operators are working on enriching real-time passenger information systems by providing passengers with train loads in addition to train schedules. This information can allow passengers to better plan their daily trips, which can improve overall comfort and avoid overcrowding on trains.

Moreover, such forecasting can be used by public transport authorities and transport operators either to enrich public transport route planning or to improve the synchronization of train traffic with passenger flows. Transport operators will increasingly need to assess and predict network passenger load to improve train regulation processes and service quality levels.

The development of smart technologies and the rapid growth in data storage abilities have increased the availability of massive transport data, such as passenger affluence, train load, real-time train schedules and so forth. This increase in data contributes to leveraging the developments in data mining and machine learning approaches for processing such spatio-temporal data to extract valuable information with the aim of providing better services to passengers or to match the transport supply with the demand. This paper addresses the forecasting of train load at a railway station considering a historical dataset that includes two data sources: train load data and automatic vehicle location. The latter source contains all information related to the train operation (delay, time of arrival/departure of vehicles and so on). Most of the prediction problems in this domain address the prediction of passenger affluence at an aggregated level (per 15 minutes or 30 minutes time horizon)[1][2][3]. In contrast to these studies, we focus here on the prediction at the non-aggregated level, taking into account real-time train schedules. This induces variability in the time step of the time series that we should predict. Furthermore, the prediction model has to take contextual factors impacting train load into account, such as calendar information (day, time, holiday and so forth) and train operation.

We address this prediction task as a multi-step short-term forecasting problem on irregularly structured times series influenced by several contextual factors. We work at the station level for each passage of train, which involves a temporal variability, making difficult the application of techniques that usually exploit structure regularity of time series. To handle these specificities, we rely on the abstraction capabilities of neural networks linked to the concept of representation learning [4]. The underlying idea is to build a mobility representation of our known influential factors. The model takes the form of an encoder-predictor neural network architecture associated with representation learning on contextual factors. It aims to predict the train load of the next trains of a station from the values of the last trains and all the contextual features characterizing these trains.

This paper is organized as follows. Section 2 is devoted to related work conducted on prediction models dedicated to public transport demand forecasting and on some main works on deep neural networks. In Section 3, we detail the case study that we consider and our dataset. Section 4 presents the proposed methodology formulated in a general framework. The evaluation of the proposed model is then conducted in Section 5 through different experiments, aiming to compare the performance of the proposed model to those provided by four baseline models on the one hand and to illustrate the learning representation space of the network on the other hand. Section 6 concludes the paper.

1.1 Related work

Numerous studies have addressed the mining of large-scale mobility data for exploration, clustering or prediction purposes. Depending on the available data, on the scale of analysis and on the targeted goal, different types of methodologies can be distinguished. For example, the authors in [5] propose predicting crowding levels from automated fare collection data by using simple techniques based on historic aggregates. Models proposed in [6] take the form of an neural network architecture using feature engineering capturing daily and monthly trends to perform daily passenger demand forecasting. The study conducted in [7] explores the viability of building a fare-recommendation system for public transport to avoid incorrect fares. For this purpose, the authors propose a two-step approach: predicting future travel habits and then matching travel habits to fares.

Extensive research on predictive models in the transport domain has been conducted. Most of the works address the prediction of passenger flows. In [8], the authors propose a deep neural network model to forecast passenger demand related to an on-demand ride service station. The combination of convolutional layers with LSTM layers allows taking spatial, temporal and exogenous dependencies into account. An LSTM recurrent neural network was proposed in [1] to address the short-term forecasting of passenger flows in a transport network. The authors in [9] work on short-term subway ridership prediction by means of a gradient boosting decision tree model. On the basis of smart card data, the authors build a prediction model using both temporal features (time and calendar) and historical data related not only to subway activities but also to bus transfer activities. Following the same line of research, the authors in [10] examine causal relationships between the adjacent flows on a public transport station with transport service features. The proposed methodology, which is based on a dynamic Bayesian network, allows highlighting such causalities and performing the prediction. Recently, the authors in [11] formalized the problem of tram load passenger prediction as a classification task, where the passenger load is labelled into different classes depending on the percentage of occupied seats. Once this labelling is performed, the authors build classical machine learning classifiers, such as k-nearest-neighbours, multi-layer perceptron, grading boosting decision tree and random forest, to predict the level of crowding in the transport. Here, temporal and historical data were used as model inputs.

Advanced machine learning models, particularly deep learning models, have also recently been proposed to address forecasting problems. Recurrent neural networks [12] are potential tools capable of capturing the dynamics of the time series. These models have even been extended to handle regular spatio-temporal data by [13] with a convolutional LSTM for weather prediction. In parallel, early research work has been accomplished by [14]. The authors propose a recurrent neural network encoder-decoder for a statistical machine translation system. This model is capable of capturing both semantic and syntactic structures of phrases. Considering the spatio-temporal dependencies in the forecasting, [3] proposes a dynamical spatio-temporal neural network for forecasting the time series of spatial processes. The idea investigated in this model is to learn both

temporal and spatial dependencies between the series to be predicted through a combined use of a latent embedding structured by the temporal dynamics of the series and a decoder mechanism to make the prediction. In [2], the authors work on forecasting the flow of crowds in all regions of a city by means of a deep-learning-based model. Historical trajectory data, weather and events are used to build the model. Residual neural networks and a parametric fusion mechanism are employed to design the forecasting model of crowd traffic.

Focusing on the transport domain, most of the aforementioned research works propose achieving the load/affluence prediction using classical classification or regression machine learning models. These models do not allow fully exploiting the sequential structure of the time series to be predicted. Moreover, they do not consider variability in time steps of the data, as it is the case here. The flexible transportation schedule, the variability in transport demand, and the contextual factors lead to complex dynamics of the series that the model should capture.

To address the structural variability in the passenger load series and influence factors, we rely on the abstraction capabilities of deep neural network models linked to the concept of representation learning [4]. The underlying idea is to learn a meaningful representation of mobility flows taking contextual factors into account. The proposed model takes the form of an RNN encoder-decoder neural network [14] associated with the representation learning of contextual factors. It aims to predict the passenger load of the next trains at a station from measures of the last trains and all the contextual features characterizing all of these trains at the same station. Note that the experiments will be performed on a real dataset collected over a long period (one and a half years), which leads to a robust evaluation of the models. Before detailing the proposed model, the next section will describe the real dataset used in the experiments.

2 Data description

Our study focuses on a dataset collected from a French railway line that serves approximately fifty stations located in the northern area of suburban Paris. The railway line carries approximately 250,000 passengers daily. The dataset covers a period of 18 months from January 2015 to June 2016 on 40 stations for daytime exploitation from 5 am to 2 am of the next day. It includes both timetable information and count data of passengers boarding and alighting at each station collected by radar sensors on trains (2000000 records covering 86% of trains). These heterogeneous sources of data that have been enriched with calendar information enable us to reconstruct the passenger load on each train departing from a station.

The main goal of this study concerns the forecasting of univariate train load series for each station. To have an idea of the time series to be predicted, Figure 1 shows an example of weekday and weekend daily train passenger load profiles collected from two stations. The suburb station accounts for approximately 22000 train stops with a particularity low train frequency and few train routes that serve this station. Conversely, the inner-city station accounts for approximately

84000 train stops with an important train frequency and multiple train routes that serve the station. Figure 1 provides insights on the forecasting problem to be solved and highlights the particularities of our dataset, namely:

- A variable sampling period due to the train timetables and railway operation. Each station has its own train frequency evolution.
- A specific temporal behaviour of each time series, which was found to be linked to the spatial location of public transport stations and geographical aspects of the city (population & employment densities, leisure and so forth).
- Train load series are impacted by calendar factors such as the type of day (weekday or weekend), holiday, public holiday and so on.
- Train load series are also impacted by train characteristics that are closely linked to their services (multi-destination line, various train services).

In addition to these contextual factors, public transport demand and therefore train load passengers can also be impacted by events (social, cultural, sport and so forth). The time series forecasting model has to face all the temporal, spatial and exogenous factors listed above. In this paper, a prediction model will be built for each railway station. The next section details the methodology developed to achieve this prediction task.

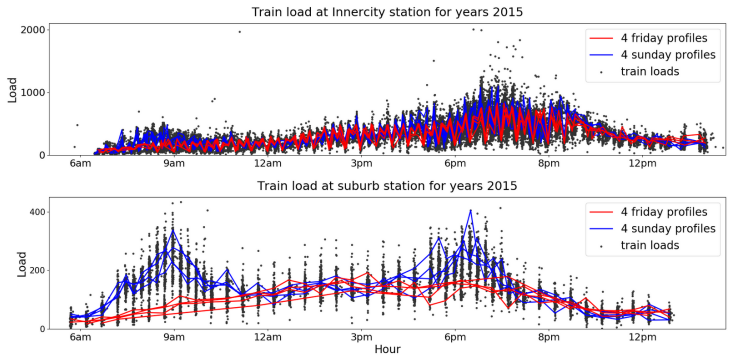

Fig. 1. Train loads in year 2015 per hour on suburb and inner-city stations

3 LSTM Encoder-Predictor

In this section we formalize the application of the recurrent encoder-predictor architecture to our particular structural constraints: a train sequence with variable time steps and heterogeneous attributes. Let $(y_1,..,y_t)$ denote the sequence to be predicted, where y_t corresponds to a passenger train load in the same station, t referring to the trains arrival order. It is assumed that each realization y_t is associated with an observation S_t which includes contextual features e_t and past measures m_t. We have tackled the issue of the variability of the time between

consecutive trains by encoding it as a contextual feature associated with specific coefficients in the model. We also use the notation y_I, S_I, e_I and m_I to designate the subsequences $(y_t)_{t\in I}, (S_t)_{t\in I}, (e_t)_{t\in I}, (m_t)_{t\in I}$ with $I \subset [1;T]$. Given a time window $W_i = [i-k, i+k']$ composed of a past horizon $P_i = [i-k, i[$ and a future horizon $F_i = [i, i+k']$, the goal of our multi-step forecasting approach is to infer a realization on the horizon y_{F_i} from information available on S_{W_i} as shown in Figure 2. Table 1 summarizes the notation used in this article.

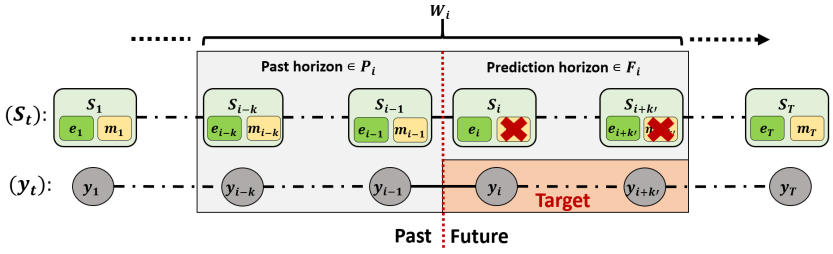

Fig. 2. Illustration of notations used in the forecasting model

This forecasting is particularly challenging because it asks to understand the laws behind realizations (y_t) taking into account the multiple influencing factors. The model must be able to dissociate the influencing factors on a structurally irregular sequence by exploiting the contextual attributes. Following the line of research on the RNN encoder-decoder proposed by [14], we propose a neural network LSTM encoder-predictor for short-term multi-step prediction including a representation learning of the contextual factors.

3.1 Method description

Given observations on a time window S_{W_i}. It aims to reconstruct the k last realizations $\hat{y}_{P_i}$ and to predict the k' next realizations $\hat{y}_{F_i}$ considering contextual information e_{P_i} and measure information m_{P_i} on past horizon and contextual information e_{F_i} on future horizon. It is a deep neural network that can be decomposed into sub-parts with specific roles. A general illustration of the proposed model is given in Figure 3. The arrangement of the different components of the LSTM are detailed in Figure 4. The sub-parts of the proposed architecture are described as follows.

$$LSTM_{EP}(X_i) = LSTM_{EP}(m_{P_i}, e_{P_i}, e_{F_i}) = (\hat{y}_{P_i}, \hat{y}_{F_i}) \tag{1}$$

Table 1. Notation and variable

Notation	
t	A time step $t \in [1, T]$
$y_1, ..., y_T$	(y_t) Realization series
$S_1, ..., S_t$	(S_t) Observation sequences
$e_1, ..., e_T$	(e_t) Sequence of feature contextual vectors
$m_1, ..., m_T$	(m_t) Sequence of feature measure vectors
	Windows
W_i	$[i-k, i+k']$: Window associated to the ith observation
P_i	$[i-k, i[$: Past horizon of window W_i
F_i	$[i, i+k']$: Prediction horizon of window W_i
X_i	$(m_{P_i}, e_{P_i}, e_{F_i})$ Input features from the window W_i
	Latent space (see subsection 3.1)
$u_1, ..., u_T$	(u_t) Contextual representation
$h_1, ..., h_T$	(h_t) Latent past dynamic
$r_1, ..., r_T$	(r_t) Latent reconstruction state
$z_1, ..., z_T$	(z_t) Latent prediction state
	Model and sub-part
$LSTM_{EP}$	Neural network model
$Fact$	MLP factoring contextual features
Enc	Recurrent encoder of past observation
Dec	Recurrent decoder of past observation.
$Pred$	Recurrent predictor of future observation
$Reconst$	MLP to reconstruct past realizations
$Predict$	MLP to predict future realizations

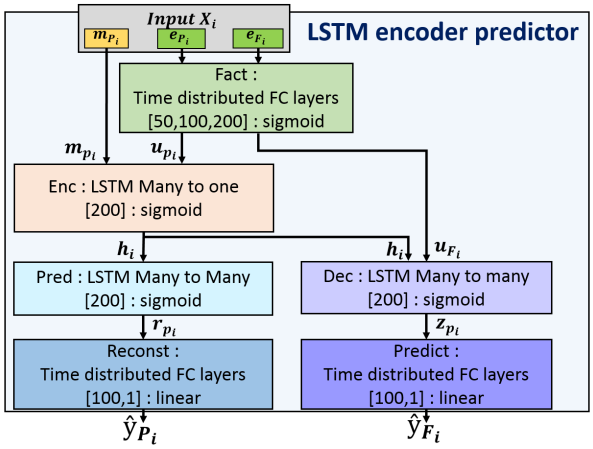

Fig. 3. General architecture of the LSTM encoder-predictor network

Fact : A context factory is dedicated to synthesize contextual features (e_t) as contextual representation (u_t). It is a preprocessing multilayer perceptron applied on each observation to regularize contextual representations.

$$Fact(e_{P_i}, e_{F_i}) = \bigoplus_{t \in (P_i \cup F_i)} Fact(e_t) = \bigoplus_{t \in (P_i \cup F_i)} u_t = (u_{P_i}, u_{F_i}) \tag{2}$$

Enc : A many-to-one LSTM 'encoder' is dedicated to capture a past latent dynamic (h_i) from the past measures m_{P_i} and the past contextual representations (u_{P_i}).

$$Enc(m_{P_i}, u_{P_i}) = h_i \tag{3}$$

Dec : A many-to-many LSTM 'decoder' decodes recurrently latent reconstruction states r_{P_i} of past observations from latent dynamics of the past horizon (h_i). Each latent reconstruction state is then interpreted by **'Reconst'**, a linear reconstruction layers that infer past observation realization. From **'Reconst'** outputs we get $\hat{y}_{P_i}$. These outputs are used as an intermediate objective during the training phase to facilitate capture of past latent dynamics.

$$Dec(h_i) = r_{P_i} \tag{4}$$

$$Reconst(r_{P_i}) = \bigoplus_{t \in P_i} Reconst(r_t) = \hat{y}_{P_i} \tag{5}$$

Enc and *Dec* form an encoder-decoder structure that synthesizing the dynamics of past observations from theirs contextual and measurement features.

Pred : A many-to-many LSTM 'predictor' infers latent prediction states (z_{F_i}) of future observations from their contextual representations (u_{F_i}) taking into account the latent dynamics of the past horizon (h_i). Each latent prediction state is then interpreted by **'Predict'**, a linear prediction layers that infer future observation realization. From **'Predict'** outputs we get $\hat{y}_{F_i}$ which corresponds to the multi-step prediction aim.

$$Pred(h_i, u_{F_i}) = z_{F_i} \tag{6}$$

$$Predict(z_{F_i}) = \bigoplus_{t \in F_i} Predict(z_t) = \hat{y}_{F_i} \tag{7}$$

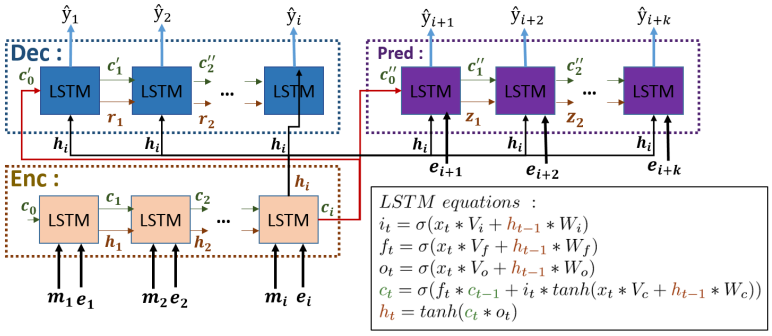

Fig. 4. Details on the layout of the LSTMs

Note that since the model is designed to address variability in the time step, this makes it straightforward to remove observations of the dataset due to missing data. Moreover, once the LSTM encoder-predictor is trained, predictions can be performed on missing data in the future horizon if we are able to reconstruct contextual information.

3.2 Optimization

A deep neural network is trained through end-to-end gradient back-propagation by minimizing the following loss function:

$$\begin{aligned} \mathcal{L}(\theta) &= \alpha_p * \sum_{t \in P_i} ||y_t - \hat{y}_t||^2 + \alpha_f * \sum_{t \in F_i} ||y_t - \hat{y}_t||^2, \\ With\ \theta &= (\theta_{Fact}, \theta_{Enc}, \theta_{Dec}, \theta_{Pred}, \theta_{Reconst}, \theta_{Predict}). \end{aligned} \tag{8}$$

The first term measures the ability of the model to reconstruct the past observations from the latent past dynamics. It is an intermediate objective that facilitates the learning of the past dynamics. The second term measures the prediction ability of the model. Hyper-Parameters α_p and α_f are the weights of the reconstruction and prediction objectives.

For the learning phase, we realize mini-batch optimization thanks to a *Nadam* optimizer [15]. Two gradients (prediction and reconstruction) are propagated from their output layers (Predict and Reconst) to the upstream layers towards the context factory through LSTM layers. The encoder-predictor is implemented based on the *TensorFlow* [16] environment and *Keras* [17] as a library and high-level neural network API.

The parameters have been chosen empirically after several experiments based on model performance and learning convergence. *Fact* is composed of 3 dense layers of size [50, 100, 200] with a sigmoid activation function for a total of 27000 parameters. *Enc*, *Dec*, and *Pred* are 3 LSTM layers of size 200 with a sigmoid activation function for a total of 880000 parameters. *Reconst* and *Predict* are

composed of 2 dense layers of size [100, 1] with a linear activation function for a total of 40000 parameters. The whole neural network has approximately 900,000 parameters.

Training is empirically realized on a batch of size 128 on several thousand iterations, which takes few hours on one standard GPU card depending on the dataset and time depth. Further work on the choice of parameters is required to improve the convergence.

4 Experimental results and discussion

For the experimental part, we evaluate two models for train load forecasting, each one on a given dataset respectively related to a station situated in the suburb and in the inner city. The datasets concern the period from January 2015 to June 2016, and they are split into training (year 2015 $\approx 66\%$) and test sets (year 2016 $\approx 33\%$). Both models use several contextual information e_t as long-term (calendar) features:

- Day of the year (8-dimensional): Position of the day in the year (365 possible values) encoded by cosine and sine of (2 $\times$4) frequencies.
- Day type (8-dimensional): Position of the day in the week with an additional attribute if the day corresponds to a holiday or not.
- Minutes (8-dimensional): Minute of the day (1440 possible values) encoded by cosine and sine of (2$\times$4) frequencies.
- Train services (8-dimensional): Feature related to the train routes that serve the considered station.

Moreover, we also consider short-term features m_t by considering a lag window that ranges between 1 to 6 past observations:

- Delay (1-dimensional): Difference in minutes between the real and the theoretical schedule of the train at the station.
- Load (6-dimensional): Number of passenger on the train for each of the last 6 passages at the considered station.
- Board (6-dimensional): Numbers of boarded passengers for each of the last 6 train passages at the station.
- Alight (6-dimensional): Numbers of alighted passengers for each of the last 6 train passages at the station.

4.1 Compared Forecasting Models

We compare on both suburb and inner-city stations different classical and machine learning models that serve as baselines for our LSTM EP:

- **Last Value (LV)**: It is the simplest forecasting that consists of forwarding the last observed load on the train to the next one at the same station.
- **Contextual Average (CA)**: It consists of using of the average load of trains that are committed on the same day type and time slice.

- **Gradient Boosting (XGB)** [18]: A regressor model that produces a prediction model in the form of an ensemble of weak decision trees. Two models are proposed depending on the input features:
  - **XGB LT**: Model that is trained based on long-term features e_t.
  - **XGB ST**: Model trained on both long e_t and short-term m_t features.
- **LSTM**: Basic recurrent network using both types of features.
- **LSTM EP**: The proposed RNN trained on context representation using both types of features.

The best parameters of XGB have been selected with a grid search procedure in conjunction with cross validation for $K= 3$ as the number of folds.

We evaluate the performance of models on each time step by root mean square error (RMSE) and weighted absolute percentage error (WAPE) measures:

$$WAPE\ score : \frac{\sum_t ||y_t - \hat{y}_t||}{\bar{y}} \tag{9}$$

WAPE is a derivative of the MAE that can be interpreted as the percentage of the overall error compared to the average value of the actual observation.

4.2 Forecasting Results

The evaluation of the forecasting models is conducted by making comparisons based on performance metrics. These metrics are expressed in terms of RMSE and WAPE and are given for both the training and the test phases for the suburb and inner-city stations. The five models defined in section 4.1, namely, the basic last value (LV), the contextual average (CA), the gradient boosting (XGB) short term (ST), long term (LT) and the LSTM-EP are compared. The errors obtained for both the training and test sets are given in Table 2.

The results show that advanced models (XGB, LSTM) outperform the LV and CA models. This performance improvement could mainly be explained by the fact that the XGB and LSTM models have better generalisation abilities and are able to fit more complex models than LV and CA, which simply predict by forwarding the last observed value or averaging historical data. The basic LSTM performs less in the inner-city station compared to the suburb station. This can be explained by its difficulties to deal with the heterogeneity in term of train services with that kind of station. Overall, LSTM EP leads to the best results since it is able to capture in better way the underlying dynamics of the temporal irregularity related to the heterogeneity of train services by means of its encoder-decoder component.

Table 2. Model performance on both studied stations

Model	Suburb			Inner city	
	WAPE	RMSE		WAPE	RMSE
LV	17.9	35.8	Train score	41.9	186.7
CA	13.7	28.7		14.2	73.1
XGB LT	8.4	17.2		8.3	44.75
XGB ST	7.5	15.1		8.2	43.5
LSTM	11.5	24.3		8.9	51.5
LSTM EP	10.7	22.1		10.9	57.7
LV	24.1	47.2	Test score	46.9	205.0
CA	19.0	40.0		18.5	96.5
XGB LT	18.8	38.9		13.4	76.0
XGB ST	16.8	35.7		12.7	73.0
LSTM	16.2	34.0		13.7	75.3
LSTM EP	**16.0**	**33.8**		**12.9**	**72.4**

Looking at the prediction error of the LSTM EP according to the load to predict (see Figure 5), we observe that errors are increasing according to the load. The model tends to slightly overestimate weakly loaded trains and greatly underestimate the highly loaded trains. Heavily loaded trains are rare and present contextual information similar to many less loaded trains, which explains the difficulty of the model in predicting large loads. To remedy this problem, provisioning features to distinguish these trains appears to be necessary. One could imagine indicators related to the disturbance of the network and the known presence of events near the station.

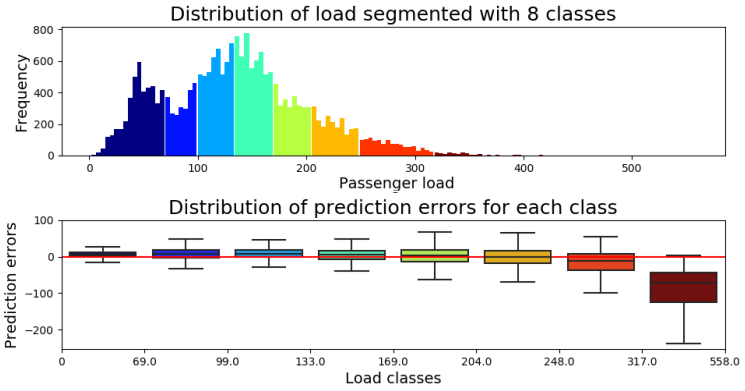

Fig. 5. Prediction errors depend on load class for suburb station

As shown in Figure 6, the model makes errors of the same order of magnitude for weekdays and weekends with different difficulties. The variance of the error

over a time slot is correlated to the importance of the load. On weekdays, we observe larger errors in the morning and afternoon peak hour linked to the strong variance and high load. The model makes average errors in the middle of the day and low errors in the morning and evening. On weekends, except in the morning, a relatively similar error variance is observed with a maximum at noon and in the middle of the afternoon. We also observe that the model has more trouble predicting weekend evenings than weekday evenings.

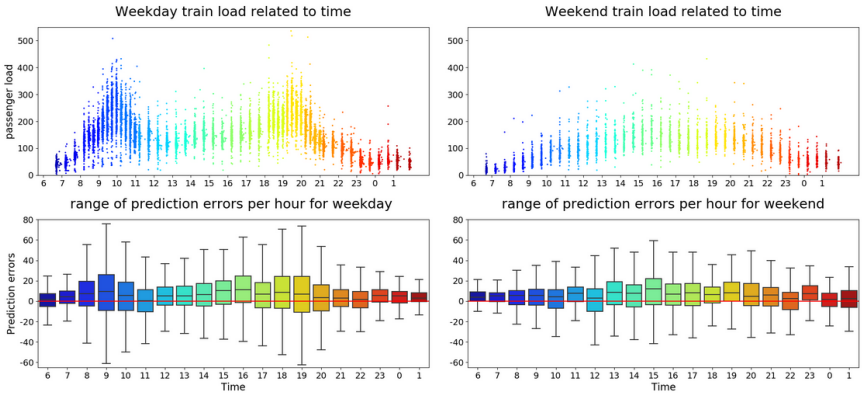

Fig. 6. Prediction errors depend on hour class for suburb station

When we examine the performance of the models on multi-step temporal prediction, the LSTM EP outperforms XGB and basic LSTM for the next 6 time steps (Table 3 and Table 4). These time steps correspond to the train passages at the station and range between 14 to 182 minutes for the suburb station, whereas it ranges between 2 to 61 minutes when considering the inner-city station since the train passages are more frequent.

Table 3. RMSE test score of the suburb station for the multi-step forecasting models

Model	t+1	t+2	t+3	t+4	t+5	t+6
Time interval*	14-32	29-62	44-92	59-122	75-152	90-182
XGB LT	38.9	38.9	38.9	38.9	38.9	38.9
XGB ST	35.7	36.6	36.7	36.7	37.6	38.1
LSTM	34.0	34.4	34.8	35.5	36.3	36.9
LSTM EP	**33.8**	**34.0**	**34.1**	**34.4**	**34.7**	**34.9**

Table 4. RMSE test score on the inner-city station for the multi-step forecasting

Model	t+1	t+2	t+3	t+4	t+5	t+6
Time interval*	2-13	5-23	9-31	12-43	15-53	18-61
XGB LT	76.0	76.0	76.0	76.0	76.0	76.0
XGB ST	73.0	72.8	73.3	73.8	73.4	73.5
LSTM	75.3	75.4	80.2	83.9	90.5	92.9
LSTM EP	**72.4**	**72.1**	**72.1**	**72.2**	**72.6**	**72.8**

The 5th and 90th percentiles of the time interval in the passage of trains at the time T and T+n

Note that these performances are obtained with a single model that simultaneously predicts the load at all the time steps for both LSTM models. The XGB LT is time-step invariant since it only considers long term features. For XGB ST, we have as many models as the considered time steps. The performance of short-term models are slightly degraded when we move forward in time, excepted for basic LSTM in the case of inner city station, where we can notice a strong degradation of its performance over time steps due to the heterogeneity of train services.The LSTM EP shows very competitive results and better robustness compared to other models for both the suburb and inner-city stations for all steps considered. This can be explained by a better understanding of contextual factors through a latent representation that helps to capture the underlying dynamics of train service at the station.

4.3 Representation learning exploration

In this section, we propose exploring the latent spaces provided by our neural network. These latent spaces correspond to the projection of contextual features and predictive state representations learned during the training phase of our train load target.

For this purpose, we first project the learned representation of the penultimate layer of our LSTM EP obtained during the training of the train load at the suburb station. Figure 7 shows the scatter plot of the dimensional reduction of the latent representation. For each sub-plot, the considered representation is projected depending on a given feature (day, hour, month, train load). Each colour block represents a common category of the handled feature. Dimensionality reduction is performed by preserving large pairwise distances between the points with the help of principal component analysis (PCA) to reduce 200-dimensional data into 50 dimensions, followed by a T-distributed stochastic neighbour embedding (tSNE) to reduce the dimensionality from 50 to 2 or 3 dimensions. As shown in Figure 7, the contextual representation (u_t) is strongly structured according to different levels depending on each contextual feature. Each point of the obtained structure reflects a train passage, and depending on the gradient of the underlying structure, we can highlight the profile of train service during weekday and weekend (see Figure 7.a), the frequency of service with a distinc-

tion between the rush and off-peak periods (see Figure 7.b), the seasonality (see Figure 7.c) and the load range (see Figure 7.d).

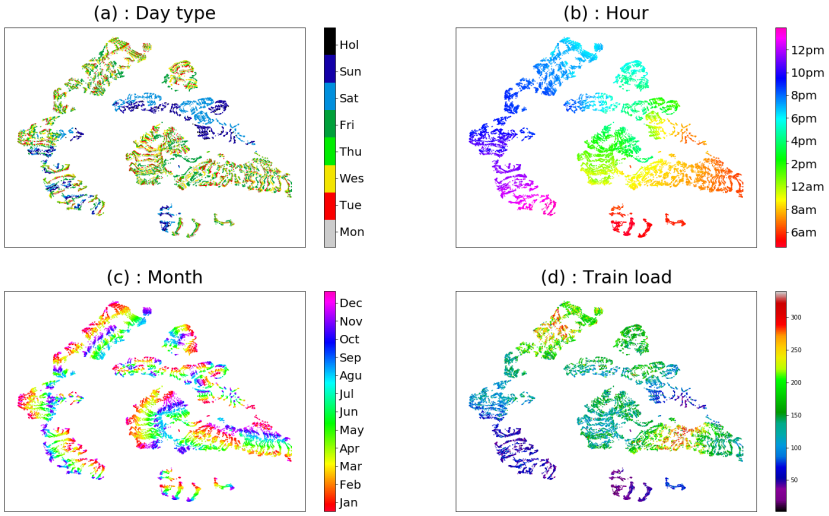

Fig. 7. Latent representation (u_t) according to contextual features obtained after dimensionality reduction (PCA + tSNE)

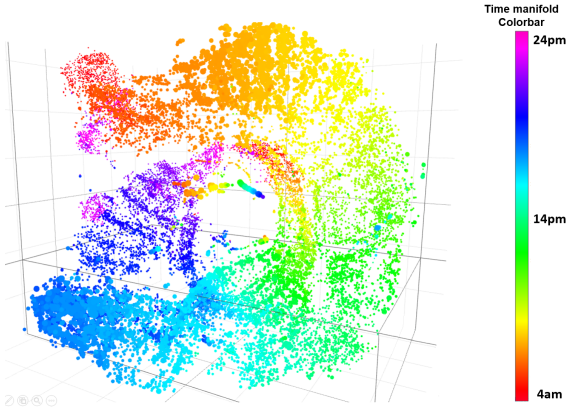

Fig. 8. Predictive latent space (z_t) after applying 3-dimensional reduction

When we reduce the predictive latent space (z_t) to 3 dimensions, we can observe the structure of the manifold as a spiralling band representing the daily evolution of the train load (Figure 8).

4.4 Conclusion

In this paper, we investigated train load forecasting on both single- and multi-step time horizons. The single step aims to predict the load for the next train at the station, whereas the goal of multi-step is to predict ahead of time the load for further train passages. The forecasting problem is particularly challenging since we have variability in train services and several contextual factors that could have a major impact on the time series to be predicted. For this purpose, we proposed a model called LSTM EP based on an encoder-decoder combined with a representation learning of contextual factors. This network has the particularity of being able to learn a contextual representation from contextual features, to capture the latent past dynamics through the underlying sub-structure of the encoder-decoder, and then forecast the forthcoming dynamics with the help of the predictive layer. The representation learning is a key value for the proposed architecture; it contributes to understanding the features and representation of the underlying dataset, which is essential for selecting the best neural network architecture for this prediction task.

The obtained results have demonstrated the potential of the LSTM encoder-predictor to address the short-term prediction of train load at stations compared to the Gradient Boosting model and basic LSTM. We evaluated the performance of the proposed model on two real datasets related to suburb and inner-city stations for single and multi-step forecasting horizons of the train load. On both configurations, the LSTM EP outperforms the LSTM, XGB and baseline models by maintaining robustness in the quality of the forecasts throughout the time horizon.

Future research should investigate the exploration of the learned representation. In particular, it would be interesting to investigate the ability of the predictive latent space to characterize abnormal situations, such as disturbances and traffic anomalies.

References

1. F. Toqué, E. Côme, M. K. El Mahrsi, and L. Oukhellou, "Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks," *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1071–1076, 2016.
2. J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for city-wide crowd flows prediction.," *AAAI Association for the Advancement of Artificial Intelligence*, pp. 1655–1661, 2017.
3. A. Ziat, E. Delasalles, L. Denoyer, and P. Gallinari, "Spatio-temporal neural networks for space-time series forecasting and relations discovery," *IEEE International Conference on Data Mining ICDM*, pp. 705–714, 2017.

4. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
5. I. Ceapa, C. Smith, and L. Capra, "Avoiding the crowds: Understanding tube station congestion patterns from trip data," *ACM SIGKDD International Workshop on Urban Computing*, pp. 134–141, 2012.
6. T.-H. Tsai, C.-K. Lee, and C.-H. Wei, "Neural network based temporal feature models for short-term railway passenger demand forecasting," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3728–3736, 2009.
7. N. Lathia and L. Capra, "Mining mobility data to minimise travellers' spending on public transport," *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1181–1189, 2011.
8. J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017.
9. C. Ding, D. Wang, X. Ma, and H. Li, "Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees," *Sustainability*, vol. 8, no. 11, p. 1100, 2016.
10. J. Roos, S. Bonnevay, and G. Gavin, "Short-term urban rail passenger flow forecasting: A dynamic bayesian network approach," *ICMLA International Conference on Machine Learning and Applications*, pp. 1034–1039, 2016.
11. L. Heydenrijk-Ottens, V. Degeler, D. Luo, N. van Oort, and J. van Lint, "Supervised learning: Predicting passenger load in public transport," *CASPT Conference on Advanced Systems in Public Transport and TransitData*, 2018.
12. F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *ICANN International Conference on Artificial Neural Networks*, 1999.
13. S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *NIPS Advances in neural information processing systems*, pp. 802–810, 2015.
14. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *EMNLP Empirical Methods in Natural Language Processing*, p. 1724–1734, 2017.
15. I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning.," *ICML International Conference on Machine Learning*, vol. 28, no. 1139-1147, p. 5, 2013.
16. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," *OSDI Symposium on Operating Systems Design and Implementation*, pp. 265–283, 2016.
17. F. Chollet *et al.*, "Keras," 2015.
18. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 785–794, 2016.