EasyChair Preprint

№ 4944

A Tutorial on Network-Wide Multi-Horizon Traffic Forecasting with Deep Learning

Giovanni Buroni, Gianluca Bontempi and Karl Determe

January 30, 2021

A tutorial on network-wide multi-horizon traffic forecasting with deep learning

Giovanni Buroni
Machine Learning Group, ULB
Bruxelles, Belgium
gburoni@ulb.be

Gianluca Bontempi
Machine Learning Group, ULB
Bruxelles, Belgium
gbonte@ulb.be

Karl Determe
Bruxelles Mobilite
Bruxelles, Belgium

ABSTRACT

Traffic flow forecasting is fundamental to today's Intelligent Transportation Systems (ITS). It involves the task of learning traffic complex dynamics in order to predict future conditions. This is particularly challenging when it comes to predict the traffic status for multiple horizons into the future and at the same time for the entire transportation network. In this context deep learning models have recently shown promising results. This models can inherently capture the non-linear space-temporal correlations (ST) in traffic by taking advantage of the huge volume of data available.
In this study the authors present a LSTM encoder-decoder for multi-horizon traffic flow predictions. We adopted a direct approach in which the model simultaneously predict traffic conditions for the entire Belgian motorway transport network at each time step. The results clearly show the superiority of this model when compared with other deep learning models. In the workshop, conference attendees will learn how to process and visualize mobility data, obtain optimal features for traffic flow forecasting, build a LSTM encoder-decoder and perform predictions in an online manner.

1 INTRODUCTION

Traffic forecasting systems are crucial in modern cities. These systems can potentially provide accurate and timely information to public and private organizations by relying on data collected in real-time. These organizations can in turn take action through policy and lead to more sustainable mobility. These aspects have enormous economic, social and environmental implications. Nowadays, thanks to the development of cutting-edge technologies and advances in artificial intelligence, traffic forecasting has reached results never seen before. In particular, the advent of deep learning (DL) models allowed to take advantage of the enormous volume of mobility data to capture the complex non-linear space-temporal relationships governing road traffic. Moreover, they proved to be particularly effective when it comes to predict traffic for both multiple forecast horizons and for multiple streets. In ITS context, the advantages of DL when compared with traditional machine learning models are summarised as follow [8, 9]:

- model huge volume of space-temporal traffic data;
- perform multi-horizon road traffic predictions on large scale transportation networks;
- achieve greater forecasting accuracy;
- comply to real-time requirements characterizing online forecasting systems in ITS;

In the recent literature for multi-horizon forecasting, DL methods can be categorised into (i) iterated approaches using autoregressive models or (ii) direct methods based on sequence-to-sequence models [7]. The former approaches use one-step-ahead prediction models where each model's output is recursively fed into future inputs to obtain multi-step predictions. On contrary, direct methods (ii) are trained to explicitly generate forecasts for multiple predefined horizons in a single step. These models generally present better forecasting accuracy than the iterative-based methods. In this paper, we present a tutorial to build and train a Direct LSTM encoder-decoder model. The model performs predictions for traffic data related to the entire Belgian motorway network. The model is tested over two-weeks period in an online fashion. The results show the DL model obtained better predictive accuracy when compared to advanced seasonal persistence model and other deep learning models. The data and models code are available at *https://www.kaggle.com/giobbu/*.

2 NETWORK-WIDE FORECASTING

The aim of traffic forecasting is to predict future traffic conditions given a sequence of historical traffic observations. These observations are detected by sensors, such as GPS, radio frequency identification devices, multi-sensors, cameras and Internet technology, that monitor the traffic status of roads in real-time. In our study, at each time step t, the traffic flow is monitored at S street segments of the entire transportation network. Hence, the road network is modelled as multiple parallel time-series and represented by the matrix $X_{[i,k]}$:

$$\begin{bmatrix} x_{1,1} & \dots & x_{1,k} & \dots & x_{1,S} \\ \vdots & & \vdots & & \vdots \\ x_{i,1} & \dots & x_{i,k} & \dots & x_{i,S} \\ \vdots & & \vdots & & \vdots \\ x_{t,1} & \dots & x_{t,k} & \dots & x_{t,S} \end{bmatrix} \tag{1}$$

where $x_{i,k}$ is the value of traffic flow at time i and street segment k.
The prediction task can be seen as the process of learning a mapping function, f from X previously observed features to Z future streets' features, $f : X \rightarrow Z$.

3 MULTI-HORIZON FORECASTING

In the study, the goal is to perform multi-horizon forecasting of the traffic flow based on the current and past traffic conditions of the entire transportation network. To achieve so, a direct strategy is implemented for multivariate multi-horizon time-series forecasts as discussed in Bontempi et al. [2].
At each time step t, we obtain:

$$f : X_t \rightarrow Z_{t+1} \tag{2}$$

where Z_{t+1} is the matrix containing the multi-horizon traffic flow predictions, $(t+1, t+2, ..., t+h)$ for all S street segments, and h is the forecast horizon.

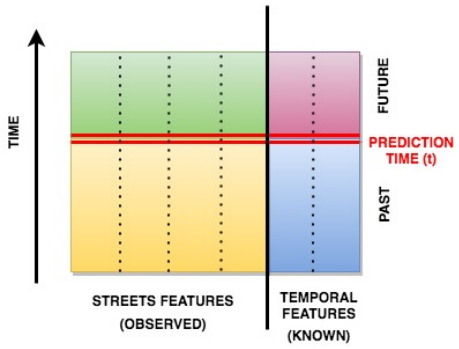

Figure 1: ***Feature engineering for multivariate multi-horizon time-series forecasts with LSTM encoder-decoder.***

4 LSTM ENCODER-DECODER

4.1 Preliminaries

4.1.1 Sequence-to-Sequence Learning. In deep learning sequence-to-sequence learning (Seq2Seq) is about training DL models to convert sequences from one domain to sequences in another domain. Multi-horizon traffic flow forecasting at transportation network scale can be seen as Seq2Seq learning problem, where the sequence of traffic flow observations from the past is learned to predict the sequence of observations in the future. The use DL models offers several advantages [4, 9]:

- natively support sequence input data (sequence of flow observations);
- directly support multiple parallel input sequences for multivariate inputs (multiple street segments);
- map input data directly to an output vector (multi-horizon predictions).

4.1.2 Long Short-Term Memory (LSTM). Long Short-Term Memory (LSTM) network is a special kind of recurrent neural network (RNN), which is capable of learning short and long-term dependencies of the input data [5] LSTM unit is composed of three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it is not important (forget gate) or to let it impact the output at the current time step (output gate). LSTMs are now widely used to deal with sequence data for learning the temporal dependency of the space-temporal data.

4.1.3 Encoder-Decoder architecture. The encoder-decoder models present a particular type of architecture particularly effective for Seq2Seq learning problem [9]. Here's how it works:

- Encoder Module: the input sequence of the encoder is fed to the module. This generates an internal state that is passed as the "context" to the decoder. Note that the outputs of the encoder are discarded.
- Decoder Module: it is trained to predict the target sequence, given the input sequence of the decoder and the memory state vectors from the encoder. The encoder's state allows the decoder to obtain information about what it is supposed to generate.

4.2 Direct LSTM encoder-decoder Architecture

At first we frame our prediction problem by diving the time-dependent input features into two categories as shown in Figure 1 : (i) the observed inputs (traffic flow of streets) which can only be retrieve at time step t and are unknown beforehand, and the time-based covariate inputs (features such as day-of-the-week at time t) which can be predetermined.
The Direct LSTM encoder-decoder (D_lstm_ED) Architecture is shown in Figure 2. As the figure shows, the encoder module takes as input all past features (both observed roads' traffic flows and known temporal features) and encode them into a latent space. This encoded "context" is then fed as internal state to the decoder module together with the known future covariates as inputs. After being trained, the model perform the forecast of the future (multiple) traffic flow values for the whole road network at each time step.

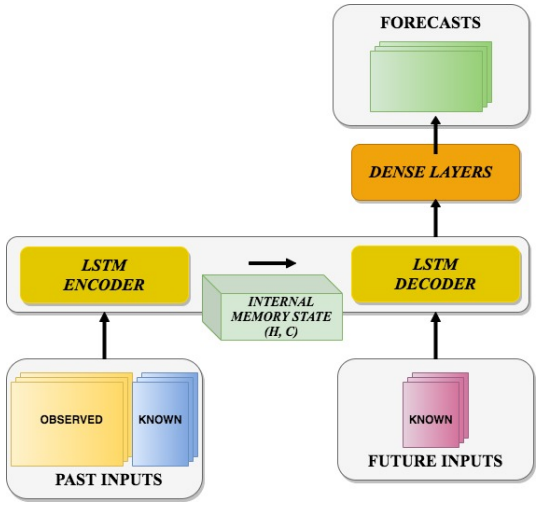

Figure 2: ***Direct LSTM encoder-decoder architecture.***

5 EXPERIMENTAL SETTINGS

5.0.1 OBU Data . As from 2016, all owners of Belgian lorries having a Maximum Authorized Mass in excess of 3.5 tonnes must pay a kilometre charge. Every road user who is not exempt from the toll must then install an On Board Unit (OBU) recording the distance that a lorry travels on Belgian public roads. Because of their value as mobility indicator, the OBU data are made available to Bruxelles Mobilite', the public administration responsible for equipment, infrastructure and mobility issues in Bruxelles-Capital Region. Each truck device sends a message approximately every 30 seconds (from 3 a.m. to 2.59 a.m. of the following day). Each OBU record contains an anonymous identifier (ID resetting every day at 3 a.m.), the timestamp, the GPS position (latitude, longitude), the speed (engine) and the direction (compass). Moreover, OBU data includes vehicle data characteristics: weight category (MAM), country code and European emission standards classification of the engine (EURO value). The large volume and the streaming nature of the OBU data required the set up of a big data platform for an efficient collection, storage and analysis [3].

5.0.2 Data Processing & Time-based Covariates. Before obtaining the matrix of Figure 1, the OBU data are processed. Firstly,

we consider two months period OBU data, from the 1st of January to the 28th of February, 2019, and the major motorways related to Belgium retrieved from `OpenStreetMap`[1] with `Module osmnx` in `Python`. Then, we filter the records with respect to their street segments location and resample with a 30 minutes interval. The trucks are considered on the road network if their location falls within the polygons area representing the street segments [2]. We took into consideration street segments with an average traffic flow higher than 10 vehicles/half-hour. The number of street segments (and consequently the number of time series analyzed) amounts to 5187 with 2832 observations (Fig. 5). Finally, starting from *Timestamp* variable, we create the following covariates:

- a *Sine* and *Cosine* transformation of the *Hours of the Day*. This will ensure that the 0 and 23 hour for example are close to each other, thus accounting the cyclical nature of the variable;
- the *Days of the Week*. Each day of week shows particular pattern of traffic flow;
- the *Working/Week-end Days*. There is a clear difference in traffic flow between the working days and the week-end days.

This features are deterministic and therefore are known in advance for future traffic flow predictions.

5.0.3 Data Preparation for DL models. In order for the DL model to learn, the sequence of observations must be transformed in the form of $\{ n_{samples}, n_{timesteps}, n_{features} \}$. Therefore, we transform X to a three dimensional tensor T_t to represent the traffic flow data, txwxM, where t denotes the total number of samples, w refers to length of sequence observations and M is the total number of features:

$$T_t = \left\{ X_1, X_2, \ldots, X_t \right\} \tag{3}$$

where X_i is the sample at time i. Thus, X_t is the matrix:

$$\begin{bmatrix} x_{t-w,1} & \cdots & x_{t-w,k} & \cdots & x_{t-w,M} \\ \vdots & & \vdots & & \vdots \\ x_{t-w+l,1} & \cdots & x_{t-w+l,k} & \cdots & x_{t-w+l,M} \\ \vdots & & \vdots & & \vdots \\ x_{t,1} & \cdots & x_{t,k} & \cdots & x_{t,M} \end{bmatrix} \tag{4}$$

The shape of each X_t is what the model expects as input for each sample.

5.0.4 Seasonal Persistence and other DL models. As a baseline to compare our DL model, we consider Seasonal Window Persistence Model (SW). Within a sliding window, observations at the same time and same day in previous three-weeks seasons are collected and the mean of those observations is returned as persisted forecast. As an example, if the data are hourly and the forecasting target is 9 a.m. on Monday, then if the window size is 1 the observation of last Monday at 9 a.m. will be returned as forecast. A window of size 2 means returning the average of the observations of the last two Mondays at the same hour.
Moreover, we compare our model with other DL models: two LSTM models that performs predictions respectively with a direct approach (D_lstm) and with an iterated approach (A_lstm) and a LSTM-based seq2seq architecture with iterated approach (A_lstm_ED) where the decoder's predictions are reinjected into the decoder's input.

5.0.5 Forecasting Evaluation and Metrics. In order to fully exploit the properties unique of real-time data, we adopt the so called *interleaved-test-then-train* evaluation scheme [1]. Each observation is used to test the model before it is used for training, and from this the forecasting accuracy is incrementally updated. Therefore, the model is always being tested on instance it has not seen. To measure the forecasting accuracy of the competing forecasting approaches we use the well-known Root Mean Squared Error, *RMSE*, and Mean Absolute Error, *MAE*. For multi-horizon forecasts these metrics are defined as follows:

$$RMSE_i = \sqrt{\frac{\sum_{i=1}^{t} \sum_{h=1}^{H} (e_{i,h})^2}{nH}} \tag{5}$$

$$MAE_i = \frac{\sum_{i=1}^{t} \sum_{h=1}^{H} |e_{i,h}|}{nH} \tag{6}$$

where $e_{i,h}$ is the error of the forecast for period i and forecast horizon h.

5.0.6 Space-Temporal Resolution and Model Setting. We forecast the *traffic flow* (vehicles/half-hour) of each street segment in the Belgian motorway network up to 6 hours ahead, forecast horizons from $h = 1$ to $h = 12$. From the whole data set we hold the last two weeks (672 observations) for testing our models. The data must be scaled to values between 0 and 1 before we can use them to train the DL model. The predictive models are initially fitted on the training set. Then, the models are incrementally updated, while subsequently adding new data from the test set as described in previous section.

Model	*avgRMSE*	*avgMAE*
Baseline	11.67	7.25
D_lstm	11.28	7.03
A_lstm	11.30	7.12
D_lstm_ED	**10.66**	**6.63**
A_lstm_ED	11.08	6.93

Table 1: *Average values of RMSE and MAE.*

5.0.7 Results. The obtained results are summarised in Table 1. The table shows the average value of *RMSE* and *MAE* metrics for both models over the testing set. The Direct LSTM encoder-decoder presents better forecasting accuracy for both metrics. Further, since traffic over the road network may be extremely heterogeneous in terms of volatility, we estimated the normalised metrics against the baseline model [6]. For example, *NRMSE* is the ratio between the *RMSE* of the predictor and the baseline *RMSE*. A ratio below 1 denotes that the considered predictor is at least more accurate than the baseline model. In Fig. 3-4, the average value of normalised metrics is computed for all forecast horizons separately. These figures highlight once again the superiority of Direct LSTM encoder-decoder when compared with other DL models especially for long-range forecast horizons.

6 OUTLINE

In the workshop, conference attendees will learn how to perform traffic flow predictions at transportation network scale by employing Direct LSTM encoder-decoder model (Fig. 5). A `Jupyter`

[1] https://www.openstreetmap.org
[2] OBU Data processing is available at *https://www.kaggle.com/giobbu/obu-data-preprocessing*

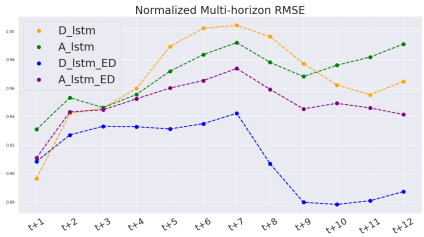

Figure 3: *NRMSE Metric.*

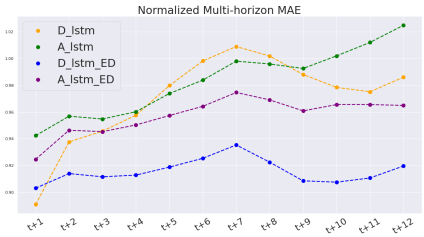

Figure 4: *NMAE Metric.*

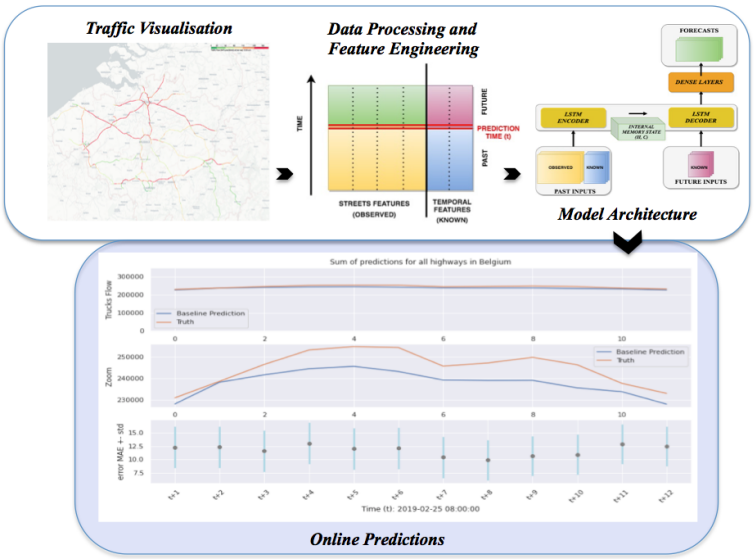

Figure 5: *Demostration Outline: from raw data to traffic flow online predictions.*

Notebook will be provided in Kaggle[3] so attendees can copy it and readily run the code and interact.

The demonstration will be organise as follow:

(1) retrieve transportation networks from OpenStreetMap[4] and OSMNX[5] in Python;
(2) process OBU data with Geopandas[6] with different time granularities and road networks;
(3) create a dataframe in Pandas[7] for traffic flow predictions and visualize it on Folium[8];
(4) add temporal features and prepare the data in Tensorflow[9];
(5) build and train a Direct LSTM encoder-decoder model in Tensorflow;
(6) perform online predictions and visualise the results with Matplotlib[10];
(7) compare the results with a Seasonal Persistence and other DL models.

ACKNOWLEDGMENTS

The authors acknowledge the support of Programme Operationnel FEDER 2014-2020 de la Region de Bruxelles Capitale and *icity.brussels*[11] for MOBIAID Project[12]. The authors are also grateful to Bruxelles Mobilite for having provided the OBU data necessary for the work.

[3]https://www.kaggle.com
[4]https://www.openstreetmap.org/
[5]https://github.com/gboeing/osmnx
[6]https://geopandas.org
[7]https://pandas.pydata.org
[8]hhttps://python-visualization.github.io/folium/
[9]https://www.tensorflow.org
[10] https://matplotlib.org

REFERENCES

[1] Albert Bifet and Richard Kirkby. 2009. Data stream mining a practical approach. (2009).
[2] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. 2012. Machine learning strategies for time series forecasting. In *European business intelligence summer school*. Springer, 62–77.
[3] Giovanni Buroni, Yann-Ael Le Borgne, Gianluca Bontempi, and Karl Determe. 2018. On-Board-Unit Data: A Big Data Platform for Scalable storage and Processing. In *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*. IEEE, 1–5.
[4] Felix A Gers, Douglas Eck, and Jürgen Schmidhuber. 2002. Applying LSTM to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*. Springer, 193–200.
[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[6] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22, 4 (2006), 679–688.
[7] Bryan Lim, Sercan O Arik, Nicolas Loeff, and Tomas Pfister. 2019. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363* (2019).
[8] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2014. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 865–873.
[9] Senzhang Wang, Jiannong Cao, and Philip Yu. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[11]https://icity.brussels
[12]https://mlg.ulb.ac.be/