TRAFFIC PREDICTION WITH CONVOLUTIONAL LONG SHORT-TERM MEMORY

Inon Peled, Francisco C. Pereira, Ole Winther

DTU, Technical University of Denmark

ABSTRACT

In recent years, Convolutional LSTM neural networks have demonstrated superior performance when applied to problems in multiple domains, including biology [1], weather forecasting [2], and speech recognition [3]. In this work, we apply this technique to the traffic domain, by measuring its accuracy in predicting speeds and flows. Our dataset comprises of 6 months of traffic information, collected from Android devices in several roads around Nørre Campus in Copenhagen, Denmark. We compare the predictive performance of Convolutional LSTM to several Recurrent Neural Network architectures which use a more "classic" Fully-Connected LSTM. The results show that for this traffic forecasting problem too, Convolutional LSTM outperforms other models.

***Keywords*—Intelligent Transportation Systems (ITS), traffic engineering and operations, short-term prediction, machine learning.**

I. INTRODUCTION

Accurate prediction of traffic conditions enables reliable planning of travel times, early detection of traffic congestion, and effective response by road practitioners. Reliable short-term traffic prediction is essential for proactive applications of Intelligent Transport Systems (ITS) [4], and is extensively used by Traffic Management Centers world-wide [5] [6].

Decades of research into short-term prediction have yielded a plethora of methods for forecasting time series data, such as the ARIMA family of models and Kalman filters. In recent years, Deep Learning techniques have also come into play, yielding successful results. A Deep Learning architecture which is often applied to time series data is Recurrent Neural Network (RNN), which in turn often employs Fully-Connected Long Short-Term Memory (FC-LSTM) units. More recently, a new type of memory unit called Convolutional LSTM (Conv-LSTM) has been published and shown to gain superior results for datasets in several domains. Conv-LSTM applies convolutions, and so is better equipped than FC-LSTM for learning patterns in local neighborhoods.

In this work, we pit FC-LSTM against Conv-LSTM within the transportation domain, in predicting speeds and flows for a network of roads in Copenhagen, Denmark. We assume that underlying our data are spatio-temporal correlations, which Conv-LSTM can take advantage of. We begin by building a simple FC-LSTM architecture, then continue to gradually enhance it to yield better predictions, compared to a Linear Regression baseline. Then, we implement a Conv-LSTM network, apply it to the same input, and show that Conv-LSTM outperforms FC-LSTM in predicting both speeds and flows.

II. DATASET

The data we use in this work consists of traffic information for several places around Nørre Campus: a campus of the University of Copenhagen. Each place falls into one of two place groups: junctions and middle-of-roads (both traffic directions). Because traffic tends to flow more freely in middle-of-roads than in junctions, we may anticipate that prediction models perform better in middle-of-roads than in junctions.

The data was collected from Android devices between 1-Jan-2015 00:00 and 29-Jun-2015 23:59, in 5 minute lags, and consists of mean speeds (km/hr) and mean flow deciles in $\{0, 1, \ldots, 9\}$. For example, if place p has flow decile 4 at lag t, then the mean flow in p during t is greater than at least 40% of all other places, and smaller than at least 50% of them.

Before modeling, we detrend speeds, and interpolate missing speeds and flows. We then use all data before June 2015 for training, while we reserve the remaining data for testing. The input to our models is in the form of feature vectors, so that for each place p and lag t, the corresponding vector comprises of speeds and/or flows in p in lags $t-1, t-2, \ldots, t-12$, namely the previous 60 minutes.

III. MODELS

For all models, we measure prediction quality through the commonly used Rooted Mean Squared Error ($RMSE$). As baseline, we use Linear Regression (LR), which we found to outperform two other common baseline models: Naive Copy (whereby the prediction is the value in the previous lag), and Historical Average. We find that LR performs essentially the same whether it learns speeds and flows together or separately.

Next, we start from a simple RNN architecture and gradually enhance it towards better prediction quality. Unless otherwise stated, all our networks use the following hyper-parameters, which we found to work best through exhaustive search: LSTM state size = 30 ; past lags = 3 ; mini-batch size = 256 ; and epochs = 100. Figures 2a and 2b provide and compare the $RMSE$ of all models.

We first experiment with "classic", Fully Connected Long Short-Term Memory (FC-LSTM). Figure 1 presents the architectures we go through to gradually improve predictive performance, as we briefly describe next. First is **LSTM Separated**, which we apply separately to speeds and to flows, and which turns out to be worse at predicting flows than LR.

Second is **LSTM Combined**, which we apply to speeds and flows together, and which outperforms both LR and LSTM Separated. This also shows that unlike LR, RNN can take advantage of processing speeds and flows together. We also experiment with adding dropout and regularization as counter-measures for overfitting, but obtain lower performance, and so leave out these additions.

Third is **LSTM Mixture**, where we first train two independent LSTM's – one for speed, another for flows – and only then combine the two. We run this architecture twice: once with state size 15 per LSTM – so that the RNN has same total memory as in the previous architectures – and once with double the state size per LSTM. The results show that LSTM Mixture performs virtually the same with or without size doubling, and is in both cases worse than LSTM Combined.

Fourth, we feed an entire place group $G \in \{\text{middle-of-roads}, \text{junctions}\}$ at once, thus allowing **LSTM Grouped** a simultaneous overview of all places. We increase the number of epochs to 150 and the state size to $30\,|G|$, and the results are slightly better than for LSTM Combined.

Finally, we create a new architecture **LSTM Conv1D**, by replacing FC-LSTM with Conv-LSTM in the LSTM Grouped architecture. We reason that two-dimensional input is problematic in our case, because the resulting grid is highly sparse, and some neighboring cells contain traffic in opposite directions. Hence we first convert the 2D map into 1D format, by ordering all places linearly in three different manners: randomly, by latitude, and by longitude. We then run LSTM Conv1D independently for each ordering, with a convolutional kernel of size 5×1.

We obtain that LSTM Conv1D outperforms LSTM Grouped – our best performing FC-LSTM model – in predicting speeds and flows, regardless of which linear ordering we apply. Furthermore, ordering by latitude or by longitude yields better performance than ordering randomly, which indicates that LSTM Conv1D indeed takes advantage of geo-spatial properties of our dataset. The best improvement is gained for ordering by longitude, as the places are more widely scattered vertically than horizontally.

IV. CONCLUSIONS

Our best performing RNN architecture is LSTM Conv1D, which uses Convolutional LSTM, rather than Fully-Connected LSTM. Compared to the baseline LR, LSTM Conv1D achieves 12% better speed RMSE in middle-of-roads, 8.5% better speed RMSE in junctions, 5.7% better flow RMSE in middle-of-roads, and 5.9% better flow RMSE in junctions.

Similarly to the baseline LR, all our RNN models yield more accurate predictions in middle-of-roads than in junctions. However, unlike LR, our RNN models successfully take advantage of combining speeds and flows at input level.

We intend to next experiment with Graph Convolutional LSTM [7], as our data is naturally structured as a graph, e.g. where places are vertices and connecting road segments are edges. We also intend to experiment with Stacked LSTM [8], which could further improve prediction accuracy.

V. REFERENCES

[1] Søren Kaae Sønderby, Casper Kaae Sønderby, Henrik Nielsen, and Ole Winther, "Convolutional lstm networks for subcellular localization of proteins," in *International Conference on Algorithms for Computational Biology*. Springer, 2015, pp. 68–80.

[2] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[3] Yu Zhang, William Chan, and Navdeep Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4845–4849.

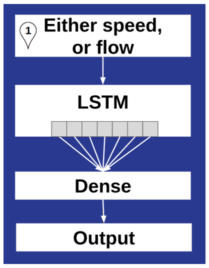
[4] F Guo, R Krishnan, and JW Polak, "Short-term traffic prediction under normal and incident conditions using singular spectrum analysis and the k-nearest neighbour method," in *IET and ITS Conference on Road Transport Information and Control (RTIC 2012)*. 2012, IET.

[5] Moshe Ben-Akiva, Michel Bierlaire, Haris Koutsopoulos, and Rabi Mishalani, "Dynamit: a simulation-based system for traffic prediction," in *DACCORD Short Term Forecasting Workshop*, 1998, pp. 1–12.

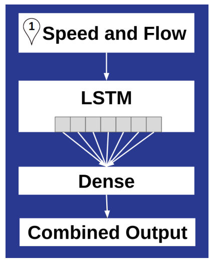
[6] Gary A Davis, Yorgos J Stephanedes, and Jeong-Gyu Kang, "Real-time traffic prediction for advanced traffic management systems: Phase i," Tech. Rep., Intelligent Transportation Systems Institute, University of Minnesota, 1995.

[7] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson, "Structured sequence modeling with graph convolutional recurrent networks," *CoRR*, vol. abs/1612.07659, 2016.

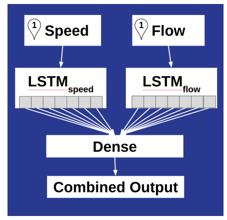
[8] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.



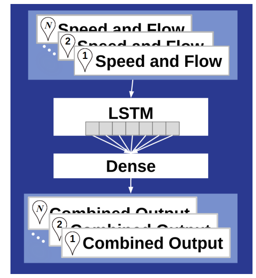
(a) LSTM Separated



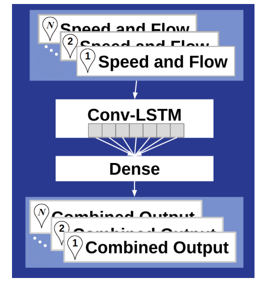
(b) LSTM Combined



(c) LSTM Mixture

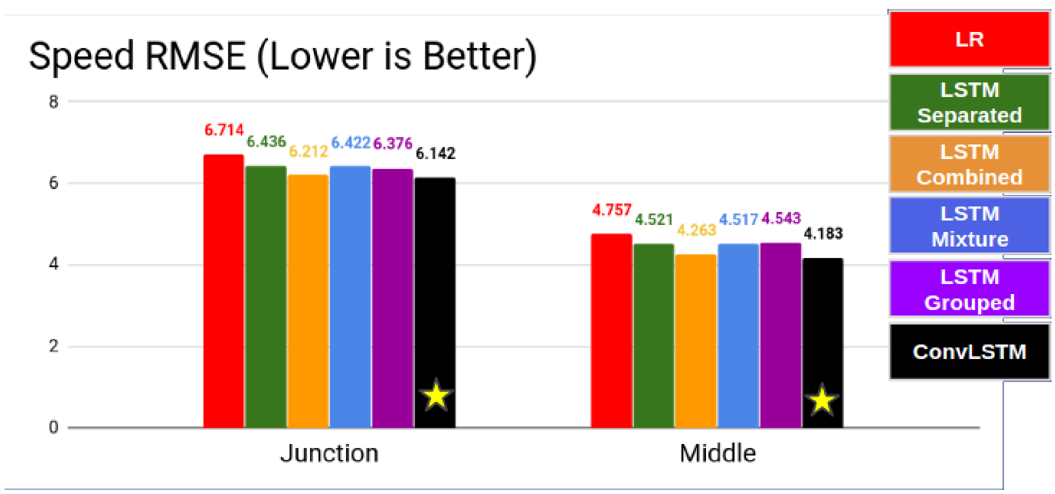


(d) LSTM Grouped

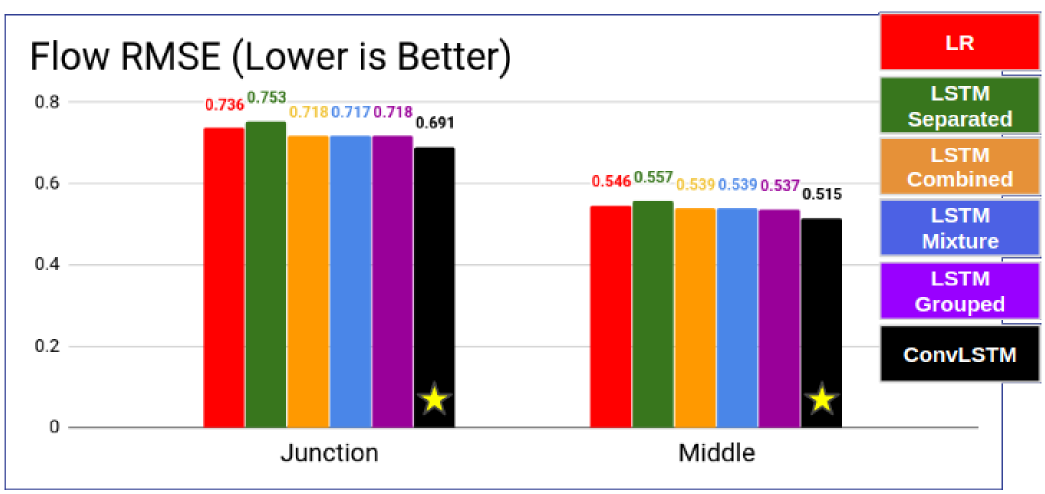


(e) LSTM Conv1D

Fig. 1: RNN architectures in our experiments.



(a) Speed RMSE



(b) Flow RMSE

Fig. 2: Comparison of RMSE for all models. A star marks the best value in each case.