A Study of Deep Learning Networks on Mobile Traffic Forecasting

Chih-Wei Huang*, Chiu-Ti Chiang*, and Qiuhui Li†
*Department of Communication Engineering, National Central University, Taoyuan, Taiwan
Email: cwhuang@ce.ncu.edu.tw, 104523047@cc.ncu.edu.tw
†College of Communication Engineering, Chongqing University, Chongqing, China
Email: 20142643@cqu.edu.cn

***Abstract*—With evolution toward the fifth generation (5G) cellular technologies, forecasting and understanding of mobile Internet traffic based on big data is the foundation to enable intelligent management features. To take full advantage of machine learning, a more comprehensive investigation on a mobile traffic dataset with the latest deep learning models is desired. Therefore, a multitask learning architecture using deep learning networks for mobile traffic forecasting is presented in this work. State-of-the-art deep learning models are studied, including 1) recurrent neural network (RNN), 2) three-dimensional convolutional neural network (3D CNN), and 3) combination of CNN and RNN (CNN-RNN). The experiments reveal that CNN and RNN can extract geographical and temporal traffic features respectively. Comparing with either deep or non-deep learning approaches, CNN-RNN is a reliable model leading in all tasks with 70 to 80% forecasting accuracy.**

***Index Terms*—Deep learning, mobile traffic forecasting, multitask learning, big data.**

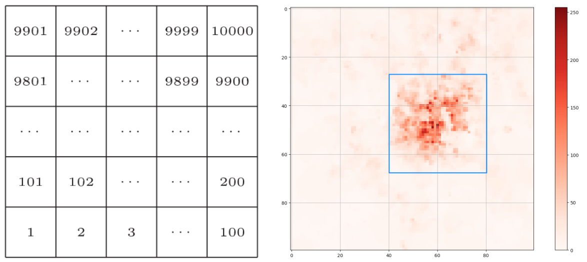
I. Introduction

With evolution toward the fifth generation (5G) cellular technologies, the communication systems are designed to be more intelligent and self-organized [1]. The self-organizing network (SON) need to adapt itself to dynamic usage patterns and take preemptive actions. Thus forecasting and understanding of the mobile traffic based on big data is the foundation to enable smart management features and highly valuable to the industry [2].

The aggregate volume of mobile Internet traffic, an essential type of core network level information, is found to have strong diurnal patterns [3]. Machine learning and time series analysis, which has been applied to a wide range of applications, can therefore be a major tool to model and predict network traffic.In [2], traffic patterns of 9000 cellular towers in a metropolitan area was analyzed. The study demonstrated that mobile traffic could be decomposed into a regular component and an unpredictable random one. After analyzing thousands of base stations and billions of records, Zhou et al. [4] classified the mobile traffic into three application types and concluded that knowledge of adjacent cell traffic could enhance the predictability of the data. Li et al. [5] proposed an application-level traffic prediction framework for cellular networks. The model used a α-stable model to capture spatial and temporal characteristics of three types of applications. Afterward, dictionary learning was implemented to refine coarse prediction results. Predicting accuracy was improved by applying customized fitting model to each type of traffic.

Deep learning-based approaches have been studied recently to discover possible feature representation from vehicular and Internet traffic flows. In [6], a stacked autoencoder model was applied to learn vehicular traffic features for prediction, based on various data collected from freeways. The prediction scheme using deep learning outperformed non-deep methods. Huang et al. [7] proposed an effective multitask learning (MTL) architecture which consists of a deep belief network for feature extraction and a multitask regression stage for supervised prediction. Oliveira et al. [8] studied algorithms based on artificial neural networks for general Internet traffic forecasting. The authors concluded that the recurrent neural network (RNN) performed better than stacked autoencoder and is suitable for time-series network traffic prediction. To take full advantage of machine learning toward intelligent network management in 5G, a more comprehensive investigation on a mobile traffic dataset with the latest deep learning models is desired.

In this work, an MTL architecture for mobile Internet traffic forecasting is presented. Tasks predicting maximum, average, and minimum traffic loads in the next epoch are jointly trained on a big dataset released by Telecom Italia [9]. We study state-of-the-art deep learning models including 1) RNN, 2) three-dimensional convolutional neural network (3D CNN), and 3) combination of CNN and RNN (CNN-RNN). The experiments reveal that CNN and RNN can extract geographical and temporal traffic features respectively. CNN-RNN is a reliable model leading in all tasks with 70 to 80% forecasting accuracy. MTL also outperform single task training (STL) with about 7% more in accuracy. To show advantages over conventional methods, deep learning approaches are further compared with the autoregressive integrated moving average (ARIMA) [10] and a non-



(a) Milan grid (b) Milan network traffic heat map

Fig. 1. Milan grid map.

deep neural network (NN) algorithm [11].

The rest of this paper is organized as follows. Section II describes details of the big dataset. Section III introduces deep learning models customized for mobile traffic forecasting tasks. The experimental results are discussed in Section IV. Conclusions are drawn in Section V.

II. Dataset Description

For data analysis with deep learning, a high-quality dataset is essential and often challenging to obtain. In this work, a multi-source dataset released by Telecom Italia in 2015 is used [9]. The dataset is one of the most comprehensive collections from an operator and also publicly available. Originally, the collection was created for a big data challenge with projects ranging from mobile networking to social applications. Provided data points include records in telecommunication, weather, news, social network, and electricity from the city of Milan and the Province of Trentino during November and December 2013. For mobile Internet traffic forecasting, we focus on telecommunication records from Milan.

Geographical grids are first defined for data recording. The city is divided into 100×100 areas with aggregated call detail record (CDR) data called Milan Grid as in Fig. 1a. Each grid has a unique square ID covering an area with the size of 235×235 meters. The data is described with standard WGS84 (EPSG:4326) GeoJson format. The telecommunications dataset contains the following information used in this work:

- *Square ID*: the identification of the square of Milan Grid.
- *Time interval*: the beginning of the time interval of the record. The end interval time can be obtained by adding 10 minutes to this value.
- *Internet traffic activity*: the number of CDRs generated during the time interval in this square id.

The Internet traffic CDR are simply referred as CDR in later sections. Nine tenths of CDRs are allocated for training. The remaining (about six days or 149 hours long) is the testing set.

III. Methodology

In this section, deep learning-based structures used for mobile Internet traffic forecasting are introduced. By observing traffic data in the last hour, the models are designed to output the maximum, average, and minimum traffic loads in the next hour. To take advantage of related outputs, multitask learning is incorporated. Three state-of-the-art deep learning models: RNN, 3D CNN, and CNN-RNN, are carefully customized for the task to capture properties of mobile traffic in various ways.

The inputs are defined as a $T \times R \times C \times F$ four-dimensional array:

- T is the number of CDRs time intervals.
- R is the number of grid rows.
- C is the number of grid columns.
- F is the number of record types.

For example, input parameters $(T, R, C, F) = (6, 15, 15, 1)$ means 6 observation time intervals, $15 \times 15 = 255$ squard grids, and 1 type of data form an input X.

A. Multitask Learning

In general, MTL learns several related tasks at the same time with the aim of sharing information between tasks. Fig. 2 shows the basic MTL architecture using RNN as the deep learning model. The network is composed of multitask regression stage and feature extraction stage. The lower stage is for features extraction and outputs values to multitask regression part. Different deep learning architectures can be used to extract features from the inputs $X_1, X_2, \ldots, X_n$ via several layers of nonlinear feature transformation. The upper stage includes flatten and regression layers and outputs values for multiple tasks $Y_1, Y_2, \ldots, Y_n$. Joint training of related tasks is likely to improve overall prediction results [7]. Therefore, three highly related tasks: maximum, average, and minimum of network traffic in the next hour, are trained together. The tasks share information in the feature extraction stage.

B. Recurrent Neural Network

To predict future network traffic in a grid, historical data have to be properly considered. The RNN model, unlike feedforward neural networks, is suitable for sequence processing by using memory in cells. Long short-term memory (LSTM) cells capable of learning long-term dependencies are adopted in the model [12]. LSTM based RNN has been demonstrated effectively on language translation [13] and speech recognition [14].

Fig. 2 shows the RNN architecture in the feature extraction stage. Variables $X_t, X_{t-1}, \ldots, X_{t-5}$ represent data of the last hour. Multiple RNN cells are stacked as a many-to-one architecture, i.e. only the cell in last time step and the top level outputs values to multitask regression stage. The number of LSTM cells in each

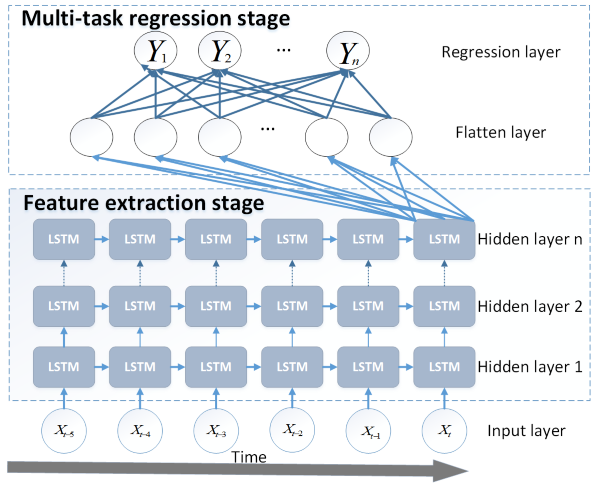


Fig. 2. An example of MTL architecture: multi-task regression on the top, RNN as feature extraction at the bottom.

hidden layer is searched from {32, 64, 128, 256, 512, 1024} and the number of layers in RNN is searched from 1 to 4.

C. 3D Convolutional Neural Network

Inspired by the digital-image-like grid structure of the data shown in Fig. 1, CNN is also considered to capture geographic-based features of mobile Internet usage. Convolution operation over the grid map keeps spatial domain information among neighbor grids. However, temporal domain features, which is essential to traffic forecasting, is largely missing in typical 2D CNN models. To perform spatial-temporal feature learning, the 3D CNN model [15] is applied with time as the third dimension. 3D CNN has been successfully utilized on human action recognition [16] and video analysis [15].

The customized 3D CNN structure is shown in Fig. 3. In the feature extraction stage, each 3D convolutional layer is immediately followed by a 3D pooling layer. Three convolutional layers and three pooling layers are constructed in the deep network. The kernel size parameters (d, k, k) and the number of kernels vary in each convolutional layer; so do the stride size and pooling policy in pooling layers. The symbol d denotes temporal depth, and k denotes spatial size. The search range is from 1 to 3 for d, from 1 to 6 for k, and from {average pooling, max pooling} for the pooling policy.

D. Combination of Convolutional and Recurrent Neural Networks

In Sec III-C, 3D CNN is introduced to extract features in the time domain. For the same reason, an architecture combining CNN and RNN is introduced and evaluated. The architecture is a deep hierarchical feature extractor with CNN learning in spatial domain and RNN in the time domain. The model has been applied for activity recognition and video description with success [17].

Fig. 4 describe the process of CNN-RNN which works by passing multiple CDR inputs (6 here: X_{t-5} to X_t) through CNN, and then to RNN stage in sequence.

In CNN the number of kernels in convolutional layers is selected from {32, 64, 128, 256, 512, 1024} to reduce complexity. The kernel and stride sizes in convolutional and pooling layers are defined as (k, k), where k is related to the spatial domain, and the values are chosen from 1 to 6. The pooling policy is searched from {average pooling, max pooling}. In RNN, LSTM cells are default cells, and the number of cells is searched from {32, 64, 128, 256, 512, 1024} in each layer. The number of layers in RNN is selected from 1 to 4.

IV. Performance Evaluation

The mobile traffic forecasting performance of aforementioned deep learning models is compared with traditional non-deep learning methods on the Telcom Italia dataset from Milan city.

A. Experimental Setup

In addition to deep learning models, non-deep learning methods of ARIMA [10] and NN [11] is also implemented for comparison. ARIMA is a commonly utilized model for time series analysis, and the NN is optimized using Levenberg-Marquardt (LM) algorithm.

CDRs are re-structured into four-dimensional arrays described in Section III. For 3D CNN and CNN-RNN, array parameters $(T, R, C, F) = (6, 15, 15, 1)$ covering a 15-by-15 area in the Milan Grid with 225 grids. In each grid, six temporal data points (one hour in total) are included with Internet traffic as the only type. The input of other prediction methods (such as RNN, ARIMA, and LM) is $(6, 1, 1, 1)$ which means there is only one grid with six CDRs during one hour.

There are three targeted network forecasting tasks in the multitask regression stage:

- Task maximum (TMAX): the maximum network traffic value out of six values in the next hour.
- Task average (TAVG): the average network traffic value of every 10 minutes in the next hour.
- Task minimum (TMIN): the minimum network traffic value out of six values in the next hour.

Output array parameters are $(T, R, C, F) = (1, 3, 3, 1)$. One data point of each grid in the 3×3 area at the center of original 15×15 grids is generated. For 5G SON applications such as data offloading, TMAX can play a more important role.

Three commonly used performance metrics: the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE) are evaluated. For output value y_i, they are defined as:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_i - \hat{y_i}| \tag{1}$$

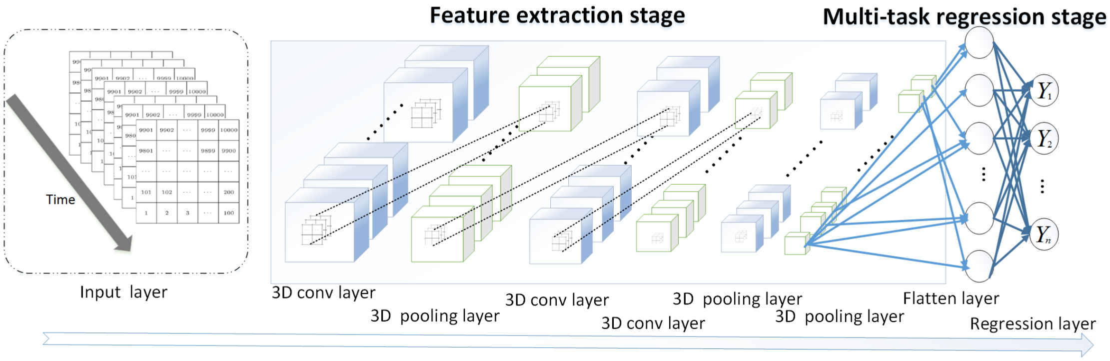


Fig. 3. 3D CNN architecture

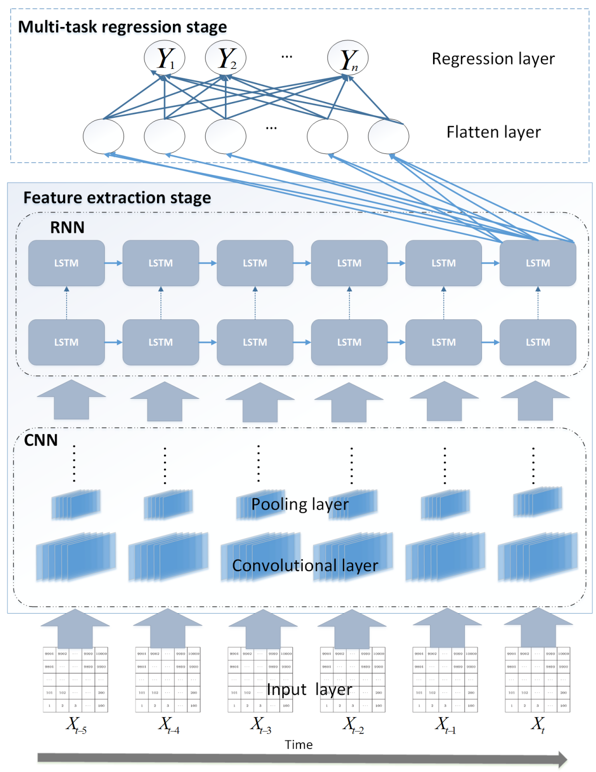


Fig. 4. Combination of CNN and RNN (CNN-RNN) architecture.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2} \tag{2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{3}$$

Mean accuracy (MA), which is derived from MAPE, is used for accuracy measurement.

$$MA = (1 - MAPE) \times 100\%. \tag{4}$$

Furthermore, deep learning models in Section III contain several parameters to be determined when building the training structure. As mentioned earlier, deep learning models with MTL are composed of two stages: multitask regression stage and features extraction stage whose architecture is different in each prediction method.

There are two fully connected layers in the multitask regression stage and grid random search is applied to choose the number of hidden units from {32, 64, 128, 256, 512, 1024} for these two layers. The searching spaces of hyper parameters in the feature extraction stage is detailed in Section III. Search results are listed in Table I for 3D CNN and Table II for CNN-RNN. Lastly, the whole experiments were performed by TensorFlow, an open source library developed by Google for deep learning [18].

TABLE I
HYPER PARAMETERS OF 3D CNN

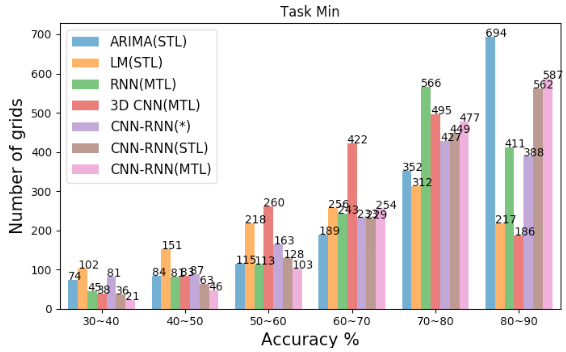
Layer	Kernels	Kernel Size	Stride Size	Pooling Policy
Conv. 1	32	3, 5, 5	2, 2, 2	
Pooling 1		2, 2, 2	1, 2, 2	Max.
Conv. 2	64	3, 5, 5	2, 2, 2	
Pooling 2		2, 2, 2	1, 1, 1	Avg.
Conv. 3	64	3, 5, 5	2, 2, 2	
Pooling 3		2, 2, 2	1, 1, 1	Avg.

TABLE II
HYPER PARAMETERS OF CNN-RNN

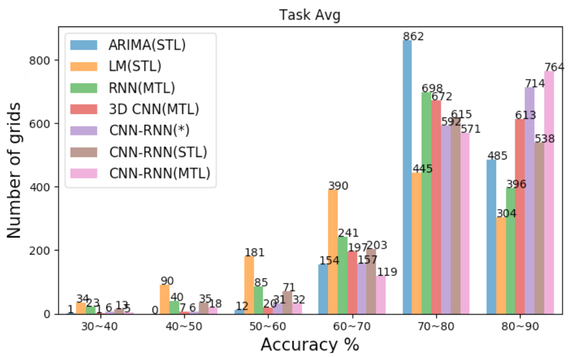
	Layer	Kernels	Kernel Size	Stride Size	Pooling Policy
CNN	Conv. 1	64	5, 5	1, 1	
	Pooling 1		2, 2	2 ,2	Max.
	Conv. 2	128	5, 5	1, 1	
	Pooling 2		2, 2	2, 2	Avg
	Conv. 3	64	5, 5	1, 1	
	Pooling 3		2, 2	1, 1	Avg.
RNN		Cells	Cell Type	Steps	
	Layer 1	256	LSTM	6	
	Layer 2	256	LSTM	6	

B. Comparison of Forecasting Approaches

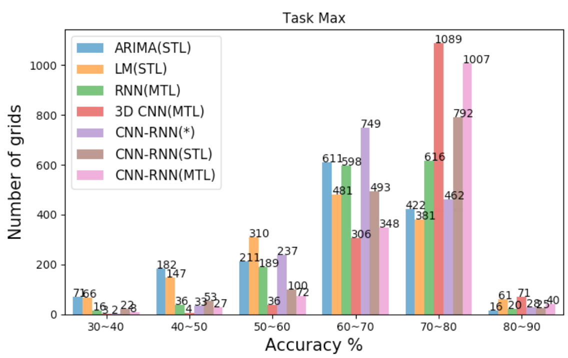
This section provides the comparison of deep learning approaches with ARIMA and LM prediction methods. The experiment is conducted on Milan downtown with



(a) Task minimum (TMIN).



(b) Task average (TAVG).



(c) Task maximum (TMAX).

Fig. 5. Distribution of accuracy in MA.

about 1500 grids according to Fig. 1b, which shows the evaluation of the network traffic activity from the CDR dataset.

Table III presents the performance of each model on targeted tasks.RNN(MTL), 3D CNN(MTL), and CNN-RNN(MTL) are performed with multitask regression learning. For comparison, STL is applied to ARIMA(STL), LM(STL), and CNN-RNN(STL). In contrast to three statistical forecasting tasks, CNN-RNN(*) is a special case where network traffic in the next hour is directly predicted for every 10 minutes without preprocessed into statistics. Analyses of STL and MTL are further discussed in Section IV-C.

In general, all approaches perform better in TAVG than TMIN and TMAX due to low variation natural of TAVG. Also, methods considering neighbor grids by CNN (3D CNN and CNN-RNN) outperform others. Among these three task, CNN-RNN(MTL) has the lowest RMSE in task TMIN and TAVG, and the next lowest RMSE in TMAX. 3D CNN(MTL) and CNN-RNN(MTL) with max pooling operations improve about 9% on TMAX over baseline models, ARIMA(STL) and RNN(MTL), in MA. CNN models with MTL perform well on TAVG and TMAX, but CNN-RNN(MTL) is significantly better than 3D CNN(MTL) in TMIN. Applying RNN in temporal domain avoids the defects.

Fig. 5 demonstrates the distribution of forecasting accuracy in MA. For TMIN, CNN-RNN and ARIMA perform significantly better between 70% to 90%. There are 1064 out of 1518 grids with MA greater than 70% using CNN-RNN(MTL), and 1046 out of 1518 grids greater than 70% using ARIMA. For TAVG, CNN-RNN, 3D CNN, and ARIMA have the most number of grids with MA 70% or higher. For TMAX, which is more valuable for traffic offloading applications, 3D CNN and CNN-RNN significantly outperform other methods at 70% to 90%. In summary, CNN-RNN(MTL) is a all-round model for traffic forecasting tasks.

C. Comparison of MTL and STL

Table IV shows advantages of MTL over STL. MTL improves the accuracy for most of the grids (about 60% of grids on TMIN, and 70% on TAVG and TMAX). The MA advantages are 6 to 7% in average. Therefore, presenting mobile traffic loads in related statistics for MTL, which shares information at the feature extraction stage, can significantly improve forecasting accuracy with deep learning architectures.

D. Effectiveness of Data Preprocessing

The output of CNN-RNN(*) directly predict the traffic load values for every 10 minutes in the next hour instead of statistics. Fig. 6 shows the performance comparison of CNN-RNN(MTL) and CNN-RNN(*) on maximum traffic prediction over a period on a grid. The CNN-RNN(MTL) model captures real maximum CDR values better than CNN-RNN(*), though the same deep learning model is applied. Overall, the CNN-RNN structure reaches 78% MA with or without preprocessing on TAVG, while the version with preprocessing, CNN-RNN(MTL), is 6% better on TMAX.

V. Conclusions

We investigate the mobile traffic forecasting problem based on CDR recording the activities in the Milan city. The evaluated deep learning structures with MTL enhance the prediction accuracy by applying CNN and RNN to capture spatial and temporal features effectively. The maximum, average, or minimum traffic loads can be better predicted for proactive management in 5G networks. The source code for this work can be found at https://github.com/IPCLab/CDRF.

TABLE III
THE PERFORMANCE OF FORECASTING APPROACHES

		ARIMA(STL)	LM(STL)	RNN(MTL)	3D CNN(MTL)	CNN-RNN(*)	CNN-RNN (STL)	CNN-RNN (MTL)
Task Min. (TMIN)	MA	67%	61%	65%	65%	60%	69%	**72%**
	MAE	22.85	32.86	24.68	24.27	27.98	21.35	19.30
	RMSE	58.18	90.96	56.84	48.81	64.65	50.4	44.37
Task Avg. (TAVG)	MA	75%	68%	67%	77%	**78%**	72%	**78%**
	MAE	24.34	33.08	31.75	22.38	21.57	26.85	21.65
	RMSE	52.25	81.46	67.87	46.93	46.08	58.36	45.77
Task Max. (TMAX)	MA	63%	63%	63%	**73%**	66%	67%	72%
	MAE	49.78	56.01	50.27	37.32	47.04	44.74	38.84
	RMSE	100.85	126.36	101.3	73.42	92.99	92.32	76.42

TABLE IV
COMPARISON OF MTL AND STL

	Total grids	MTL is better (grids)	MA improvement
Task Min.	1503	881 (59%)	7%
Task Avg.	1503	1033 (69%)	7.6%
Task Max.	1498	1007 (67%)	6.2%

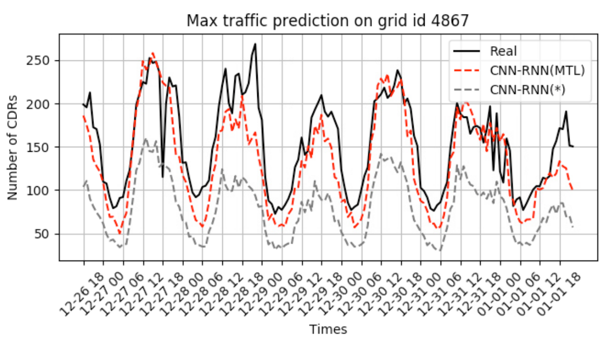


Fig. 6. Comparison of CNN-RNN(MTL) and CNN-RNN(*) on maximum traffic prediction over a period of time on a grid.

ACKNOWLEDGMENT

The research is based on work supported by the Ministry of Science and Technology (MOST) of Taiwan, under grant number MOEA 106-2221-E-008-019.

REFERENCES

[1] A. Imran and A. Zoha, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov. 2014.

[2] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach," *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 796–805, Sep. 2016.

[3] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. New York, New York, USA: ACM Press, 2011, p. 305.

[4] X. Zhou, Z. Zhao, R. Li, Yifan Zhou, and H. Zhang, "The predictability of cellular networks traffic," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, oct 2012, pp. 973–978.

[5] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, "The Learning and Prediction of Application-Level Traffic Data in Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3899–3912, jun 2017.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[7] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[8] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: a comparison between traditional and deep learning neural networks," *International Journal of Big Data Intelligence*, vol. 3, no. 1, p. 28, 2016.

[9] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the Province of Trentino." *Scientific data*, vol. 2, p. 150055, 2015.

[10] A. Adas, "Traffic models in broadband networks," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 82–89, jul 1997.

[11] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and LevenbergMarquardt Algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, Jun. 2012.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[14] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015.

[18] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/