November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

Comparison of Traffic Flow Prediction Models Based on Deep Learning

Pregya Poonia, V.K. Jain
Mody University of Science & Technology, Lakshmangarh (Sikar), Rajasthan, India
pinta22pragya@gmail.com, drvkjain72@gmail.com

Article Info
Volume 81
Page Number:2291 - 2300
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *12 December 2019*

Abstract:
Traffic flow prediction is a challenging task in Intelligent Transportation System (ITS). Accurate information of traffic flow help travelers to plan their routes wisely. It can help in reducing traffic congestion and improves efficiency. A number of traffic flow techniques exist but they fail to provide promising results that is because of their shallow learning architecture. When we compare these shallow architectures to deep learning architectures they are lack of feature learning capability. In this paper, we have introduced four deep learning architectures; LSTM,I-LSTM, RNN-LSTM and CNN-LSTM to predict the flow of traffic. These proposed models are applied on real-time traffic data collected from Jaipur, Rajasthan. It includes the comparison of all these architectures to find out the best one.

Keywords: *JSI Method, ART Tool, MSDs,Barecore Product*

I. INTRODUCTION

Anticipating the traffic stream is an unpredictable procedure that is influenced by a few parameters, for example, traffic designs, information accumulation, applied zones, and so forth the rightness of traffic stream expectation can acquire preferred position to the smart traffic the executives, it can help in improving rush hour gridlock productivity and diminishing traffic blockage. Fundamentally, stream forecast targets assessing the absolute number of vehicles given a particular district and a period interim. As indicated by Wenhao Huang et every one of the, a dependable constant traffic stream expectation should bolster: 1) advance continuous course direction; 2) authentic traffic control strategist; 3) diagnosis of these guidance and control strategist. These things can help in saving money and time, and also can help in reduction of traffic congestion and accidents[15].

Most of the current researches are focused on this area in recent years.Current prediction model involves two approaches: parametric and non-parametric [6, 8]. One of the famous parametric techniques is ARIMA (Autoregressive Integrated Moving Average)model; it provides perfect solution to some time series problems. ARIMA was also the first model which introduced to predict the traffic flow. This model was working on low volume data, but the problem started when large volume of data was given to the model [13, 14]. ARIMA model cannot tackle extreme volume of traffic data.Later on many updated versions of ARIMA came, like; seasonal ARIMA, ARIMA with Fuzzy and few more, but they all failed to process large volume of data. This leads to more researches on this area [12, 18].

Neural Networks are the second approach researchers started working on. The vast majority of the NN methodologies are structured with single layer. The purpose behind the disappointment of NN approach might be the ineffective preparing methodology with various shrouded layers [5, 10]. For different techniques earlier information of specific areas are required, they utilize this learning for highlight extraction and choice. Finally, they just figure the traffic stream of every street independently and ignore sharing knowledge among related roads [10].

In this paper, we have determined different deep architecture which learns features with limited prior information. And have made a comparison table to find out the best one suitable for traffic flow forecasting. We have taken four deep learning models;

1) LSTM (Long Short-Term Memory Networks) – it is an RNN architecture which is designed to avoid vanishing gradient problem. It can use model parameters more effectively to train large scale prediction models. LSTM model is able to predict large scale traffic flow predictions.
2) RNN-LSTM (Recurrent Neural Network) –RNN is one of the best model to perform time-series data.
3) CNN-LSTM (Convolution Neural Network) – CNN model can learn compact yet discriminative feature representation. It has powerful learning capabilities.This helps in feature extraction and prediction.
4) I-LSTM – I-LSTM is an upgraded version of LSTM algorithm. It can process both linear and non-linear datasets where as LSTM is able to process only linear data.

These are the models we are working on.

II. DEEP LEARNING ARCHITECTURE

It is very similar to human brain. As a human brain understands and consumes knowledge the same pattern is applied to a particular situation so that's where we use Deep Learning. In deep learning we are automating the entire process which a human brain does. We do not focus on visualization or data management we focus to put whole things into one system that does the entire system on an automated basis 3, 4]. Deep learning is reinforcement learning methodology. It's far more advance, far more sophisticated. It is an 'end-to-end' scenario. Deep learning networks can learn features that traditional neural networks cannot.

The core of Deep Learning is neuron. Neurons are the basic building blocks of a human brain [7, 17]. Deep networks have more number of neurons which add non-linearity to the data, more weighs to adjust, increases accuracy and give greater precision output.

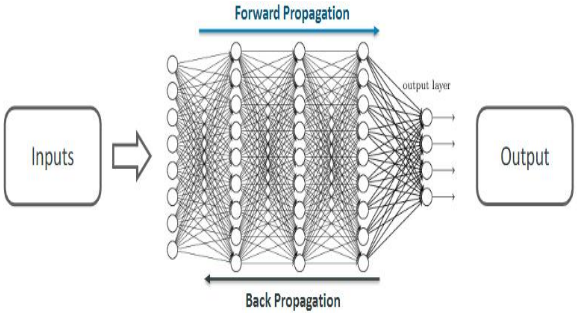


Fig1: Working of Deep Networks

Thing which makes it different from shallow architectures are:

- More neurons than previous networks.
- More complex ways of connecting layers.
- Advancement in computing power.
- Automatic feature extraction.

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

A. Deep Learning Models

In this section we will discuss about the deep leaning models we are working on.

a) Long Short-Term Memory Networks (LSTM) –

LSTM is distinctly designed to avoid the long-term dependency issue. Recalling data for extensive stretches of time is basically their default conduct. LSTM only stores relevant information, one of the best example of this is – Chatpot [1, 2].
It has following layers; 1) sigmoid; 2) tanh; 3)pre-sigmoid. Sigmoid&tanh layers help in retaining the memory.

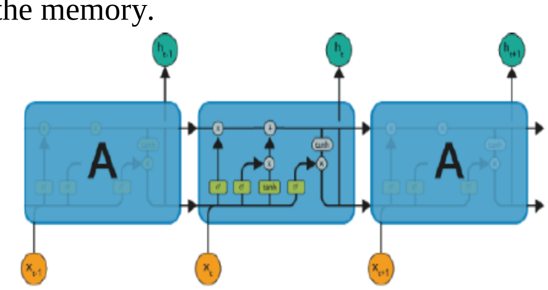


Fig2: Structure of LSTM

The best approach to LSTM is cell express, the level line experiencing the most astounding purpose of the diagram. It is a transport line which conveys all the data that is important to the past information. LSTM has the limit ro clear and add information to the telephone state, carefully constrained by structures called portals [19].

Gates are the best way to deal with on the other hand let information through. LSTM has three portals to verify and control the cell state.

Working Steps of LSTM-

The underlying stage in LSTM is to perceive that information which is not required and will be disposed of from the cell state. This conclusion is made by sigmoid layer called as overlook forget gate layer. Considering $h_{t-1}$and x_t, overlook gate layer yields a number somewhere in the range of 0 and 1 for each number in the phone state c_{t-1}. Here, 1 signifies that the information will be kept whereas in case of 0, it will be thrown away.
The equation will be like-

$$f_t = \sigma\,(w_f . [h_{t-1}, x_t] + b_f)$$

Subsequent stage is to choose which data we are going to store in the cell state.
$i_t = \sigma\,(w_i . [h_{t-1}\,, x_t] = b_i)$
$c_t = \tanh\,(w_c\ . [h_{t-1}\,, x_t] = b_c)$

In next step, we will modulate these two to create an update the state. Now c_{t-1} will be updated to c_T .

$$c_T = f_t * c_{t-1} + i_t * c_t$$

Finally, it will run a sigmoid layer which chooses what parts of the cell state we are going to yield.

$O_t = \sigma\,(w_0\ . [h_{t-1}\,, x_t] +\ b_0)$
$h_t =\ O_t * \tanh\,(c_T)$

These are the working steps of LSTM.

b) Recurrent Neural Network with Long Short-Term Memory(RNN-LSTM) –

For sequential data the lengthof the input sequences can vary from example to example. Also in sequence there will be sort and long temporal dependencies among the words, thus normal deep networks cannot be used for sequential data [11, 19]. Thus we need a network which can solve the problems we are facing in solving time-series data and RNN is the network which can process time-series data very promisingly.

TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

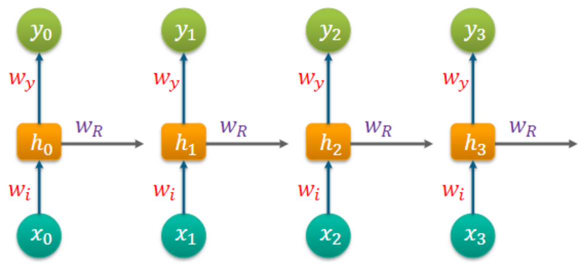


Fig3: Structure of RNN

Basic elements of a recurrent neuron are:

$h^{(t)} = g_h(w_i x^{(t)} + w_R h^{(t-1)} + b_y)$

$y^{(t)} = g_y\ (w_y h^{(t)} + b_y)$

In the above equations:

w_i = weight for x, it is the generic weight which we disclose while running a deep learning model.

w_y = weight for y.

w_R = recurrent weight, it is helpful in processing the next information. It takes it from the previous information and adds it to the next information.

$w_R h^{(t-1)}$ = it indicates the activity at previous time stamp.

Here, $h^{(t)}$ depends on both input $x^{(t)}$ and a recurrent connection with weightw_R.

Recurrent neural network uses backpropagation to train the model. And the major drawback of this algorithm is 'vanishing gradient'. One more drawback of RNN is it cannot store information for a long period of time this will affect the accuracy of the algorithm. To overcome this problem a different architecture called, Long Short-Term Memory Networks added to the RNN architecture. LSTMs are a distinctive kind of RNN, eligible of learning long-term dependencies.

c) Convolutional Neural Network with Long Short-Term Memory Networks (CNN-LSTM) –

CNN and RNN use same computational methodology of ANN, the difference is they use some different methods to calculate the data.

CNN is a kind of feed-forward neural framework wherein the accessibility structure between its neurons is awakened by the relationship of the animal visual cortex [21]. In CNN the neuron in a layer might be related with a little region of the layer before it, as opposed to all of the neurons in a totally related manner. CNN is widely used for image recognition and classification problems.

It has three types of layers:

- Convolutional Layer
- ReLU Layer
- Pooling Layer

CNN first compresses the data and then looks for features by finding rough feature matches.

1) In convolutional layer, we perform convolution of a particular feature which is the process of trying every possible position.
2) In ReLU layer, we remove every negative values from the filtered dataset and replace them with zero's.
3) In pooling layer, we shrink the dataset into a smaller size.

Fig4: CNN Architecture.

In the last step one more layer, Fully Connected Layer, is added where the actual classification happens and gives the final output.

CNN is best for image classification so to perform it on time-series data we have added the LSTM model to see how well it works on these types of datasets.

d) I-LSTM:

I-LSTM is an upgraded version of long short-term networks. It can process both linear and non-linear

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

datasets where as LSTM is only process linear data.

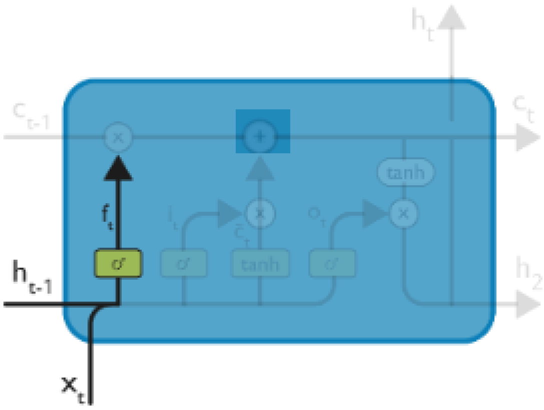


Fig5: Working of LSTM.

Working steps of I-LSTM will be the same as LSTM algorithm the only thing added is a Dropout layer which helps in processing both linear and non-linear dataset.

```
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50, return_sequences=True))
```

III. EXPERIMENTS AND RESULTS

In this present study following software's were used-

a) Anaconda – it is a free and open-source environment for Python and R programming languages for machine learning applications, data science, predictive analytics and large-scale data processing. It simplifies package management and deployment. In this study we have used Anaconda version 3.

b) Jupyter Notebook –it is a tool within anaconda, lots of Deep Learning packages are already installed in Jupyter notebook so we don't need to spend time on installation part.

c) Keras – it is a high-level neural networks API, written in python. Keras is preferred because it allows easy and fast prototyping; supports convolutional and recurrent networks; runs seamlessly on CPU and GPU. In this study we have used Keras 2.2.4 version.

A. Techniques

In this section methods and techniques used for prediction of traffic flow were discussed.

a) Adam Optimizer –from previous findings we found Adam Optimizer is the best suited method for optimization. Adam is an optimization algorithm which helps in update network weights iterative based on training data.

b) RMSE (Root Mean Square Error) – this is an error estimation function, we have used this to calculate the loss while training and testing the data. At the end, it tells how focused the information is around the line of best fit.

The formula is:

$$RMSE = \sqrt{\overline{(f-o)^2}}$$

Where-

f = forecasts (expected valued or unknown results)

o = observed values (known results).

The bar above the squared differences is the mean.

c) Scatterplot Matrices – it shows the linear correlation between multiple variables. These are **good** for determining rough linear correlations of metadata that contain continuous variables.

d) Spearman Rank Correlation –it is a nonparametric proportion of rank connection it indicates how well the connection between two factors can be depicted.Spearman's returns a value from -

TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

1 to 1, where the data must be ordinal, interval or ratio.

B. Data Description

We applied the proposed models on the data collected by MNIT, Jaipur. The work of counting vehicles on particular time intervals was assigned by the government to the civil department of MNIT College for the purpose of designing overpass to diverse the traffic flow in particular areas. The dataset we have used in our project was collected from the area of Gonar to Jagatpura. The data then aggregated into a 1-hour interval. Four months data is selected for experiment. Here we have only analyzed the data to predict traffic flow.

C. Implementation

a) LSTM –

The result of experiment is:

Epochs	Time Taken	Loss
1	6sec	0.0175
2	3sec	0.0073
3	3sec	0.0065
4	4sec	0.0059
5	5sec	0.0057

Our model has taken 5sec to count the loss of 0.0057.

❖ Difference metrics of actual value and comparison value based on working day and weekend days.

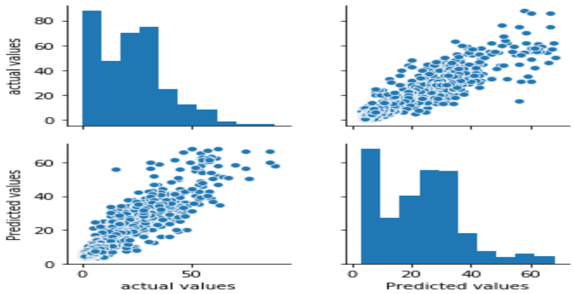
MONDAY		
Time	Actual Value	Predicted Value
7-10AM	84	72.229906
12-3PM	76	81.300301
5-8PM	96	78.323901
8-10PM	79	57.411804

SATURDAY		
Time	Actual Value	Predicted Value
7-10AM	74	69.254698
12-3PM	68	67.229470
5-8PM	41	49.001301
8-10PM	27	35.748743

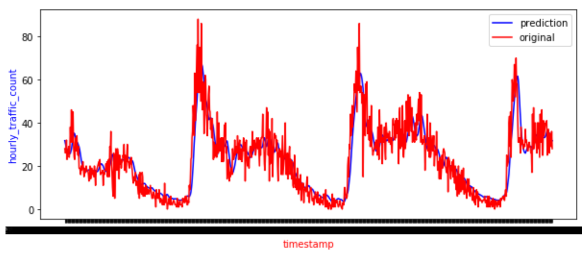
SUNDAY		
Time	Actual Value	Predicted Value
7-10AM	35	33.180220
12-3PM	31	33.327859
5-8PM	25	31.947026
8-10PM	17	20.372929

❖ Scatterplot matrices :

:



❖ Comparison graph of Actual values and Predicted values:



b) RNN-LSTM:

Result:

Epochs	Time Taken	Loss
1	4sec	0.0245
2	1sec	0.0057
3	1sec	0.0051
4	1sec	0.0050
5	1sec	0.0049

RNN-LSTM took 1sec to count the loss of 0.0049.

TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

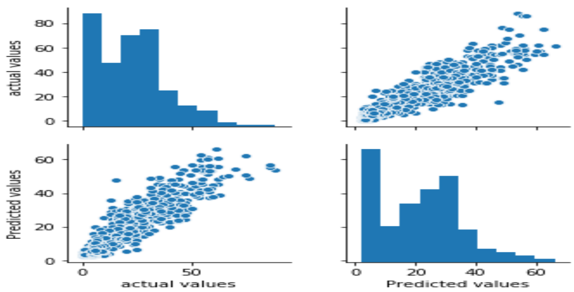
❖ Difference metrics of actual value and comparison value:

MONDAY		
Time	Actual Value	Predicted Value
7-10AM	84	63.691153
12-3PM	76	77.422329
5-8PM	96	79.513212
8-10PM	79	64.200752

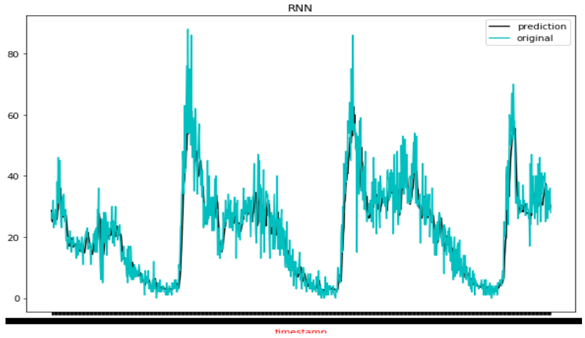
SATURDAY		
Time	Actual Value	Predicted Value
7-10AM	74	69.275020
12-3PM	68	67.610650
5-8PM	41	62.922964
8-10PM	27	35.556635

SUNDAY		
Time	Actual Value	Predicted Value
7-10AM	35	33.780220
12-3PM	31	33.885040
5-8PM	25	31.947026
8-10PM	17	20.709085

❖ Scatterplot metrics:



❖ Comparison graph of Actual values and Predicted values:



c) CNN-LSTM

Result:

Epochs	Time Taken	Loss
1	5sec	0.0517
2	2sec	0.0283
3	2sec	0.0246
4	2sec	0.0246
5	2sec	0.0263

CNN-LSTM took 2sec to count the loss of 0.0263.

❖ Difference metrics of actual value and comparison value:

MONDAY		
Time	Actual Value	Predicted Value
7-10AM	84	58.950350
12-3PM	76	61.823463
5-8PM	96	60.243929
8-10PM	79	41.219206

SATURDAY		
Time	Actual Value	Predicted Value
7-10AM	74	69.254698
12-3PM	68	67.224470
5-8PM	41	62.160420
8-10PM	27	35.556635

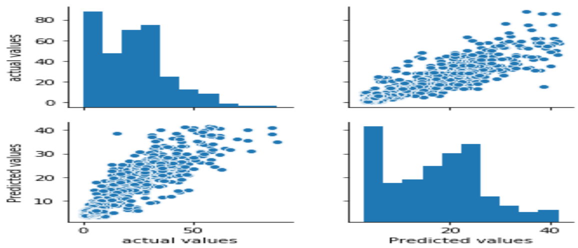
SUNDAY		
Time	Actual Value	Predicted Value
7-10AM	35	33.885040

TEST Engineering & Management

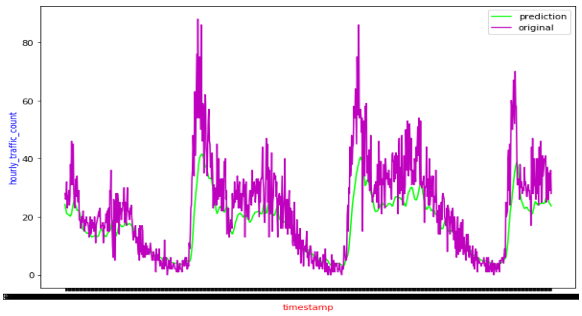
November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

12-3PM	31	33.780220
5-8PM	25	31.947026
8-10PM	17	20.372929

❖ Scatterplot metrics:



❖ Comparison graph of Actual values and Predicted values:



d) I-LSTM –

Results:

Epochs	Time Taken	Loss
1	16sec	0.0258
2	8sec	0.0121
3	8sec	0.0092
4	8sec	0.0079
5	8sec	0.0070

I-LSTM took 8sec to count the loss of 0.0070.

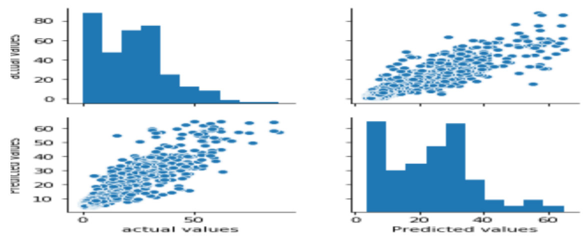
❖ Difference metrics of actual value and comparison value:

MONDAY		
Time	Actual Value	Predicted Value
7-10AM	84	68.971092
12-3PM	76	89.287811
5-8PM	96	76.382776
8-10PM	79	52.052260

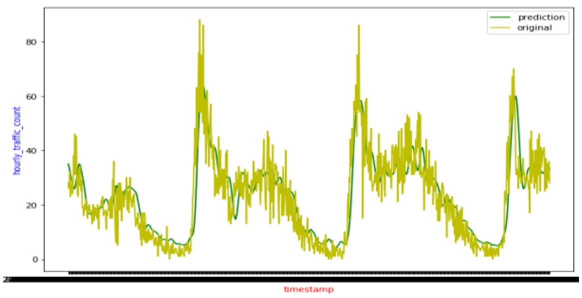
SATURDAY		
Time	Actual Value	Predicted Value
7-10AM	74	69.254698
12-3PM	68	45.229470
5-8PM	41	62.160420
8-10PM	27	35.556635

SUNDAY		
Time	Actual Value	Predicted Value
7-10AM	35	33.885040
12-3PM	31	33.780220
5-8PM	25	31.947026
8-10PM	17	20.372929

❖ Scatterplot Metrics:



❖ Comparison graph of Actual values and Predicted values:



Comparison of Experimental Results: Models	MSE	RMSE	Time-Taken	Predicted Value	Actual Value
LSTM	0.0057	7.25	5sec	31.65	28
RNN-LSTM	0.0049	6.76	1sec	30.34	28
CNN-LSTM	0.0263	10.26	2sec	23.69	28
I-LSTM	0.0070	7.89	8sec	32.92	28

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

Here, MSE is the error in training process; RMSE is the error in testing process; Time-Taken is the time a model took to predict the value and find out the loss.

From the above table we can see that RNN-LSTM is the best fitted model for traffic flow prediction.

IV. CONCLUSION

In this paper we have proposed four Deep Learning models to predict the short-term traffic flow. The dataset we have applied on this model has the time interval of 1-hour. Result shows that among all these models RNN-LSTM give promising results to predict traffic flow.

We have proposed a new model 'I-LSTM' by modifying LSTM algorithm. As we know that LSTM takes linear-datasets only, it doesn't work on non-linear datasets. The modified algorithm works on both linear and non-linear dataset.

REFERENCES

[1] Hongnix Shao and Boon-Hee Soong, "Traffic flow prediction with Long Short-Term Memory Networks (LSTMS)," 2016 IEEE Region 10 Conference (TENCON) – Proceedings of the International Conference.

[2] TjengWawanCenggoro, and Ida Siahaan, "Dynamic Bandwidth Management Based on Traffic Prediction Using Deep Long Short Term Memory,"2016 2nd International Conference on Science in Information Technology (ICSITech).

[3] YuhanJia, Jianping, and Yiman Du, "Traffic speed prediction using deep learning method," 2016 IEEE 19th International Conference on Intelligent Transportation System (ITSC), November 1-4, 2016.

[4] Joe Lemieux, and Yuan Ma, "Vehicle Speed Prediction using Deep Learning," 2015 IEEE Conference.

[5] YishengLv, YanjieDuan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang, Fellow, IEEE, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," IEEE transactions on intelligent transportation systems, vol. 16, no. 2, april 2015.

[6] Wenhao Huang, Guojie Song, Haikun Hong, and KunqingXie, "Deep Architecture for Traffic Flow Prediction: DeepBelief Networks With Multitask Learning," IEEE transactions on intelligent transportation systems, vol. 15, no. 5, october 2014.

[7] Francis Quintal Lauzon, "An introduction to deep learning," The 11th International Conference on Information Sciences, Signal Processing and their Applications: Special Sessions, 2012.

[8] Zhiheng Yu, Tieli Sun, Hongguang Sun and Fengqin Yang, "Research on Combinational Forecast Models for the Traffic Flow," Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 201686, 10 pages.

[9] KartikeyaJha, NishitaSinha, Shriniwas S. Arkatkar, and Ashoke K. Sarkar, "A comparative study on application of time series analysis for traffic forecasting in India: prospects and limitations," current science, vol. 110, no. 3, 10 february 2016.

[10] Daniel Kubeka, PawełWięceka, KonradChwasteka,"The impact of short term traffic forecasting on the effectiveness of vehicles routes planning in urban areas," XII Conference on Transport Engineering, CIT 2016, 7-9 June 2016, Valencia, Spain.

[11] Zhisheng Wang, Jun Lin, Student Member, IEEE, and Zhongfeng Wang, "Accelerating Recurrent Neural Networks: A Memory-Efficient Approach," IEEE transactions on very large scale integration (vlsi) systems, vol. 25, no. 10, october 2017.

[12] AriefKoesdwiady, Student Member, IEEE, RidhaSoua, and FakhreddineKarray, "Improving Traffic Flow Prediction With Weather Information in Connected Cars:

[13] A Deep Learning Approach," IEEE transactions on vehicular technology, vol. 65, no. 12, december 2016.

[14] S. Vasantha Kumar1 &Lelitha Vanajakshi2, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," Springer 2015.

November-December 2019
ISSN: 0193-4120 Page No. 2291 - 2300

[15] Billy M. Williams, M.ASCE1 and Lester A. Hoel, F.ASCE2, “Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical BasisAnd Empirical Results,” 10.1061/~ASCE!0733-947X~2003!129:6~664!.
[16] AshratuzZavin, Adnan Sharif, AnikaIbnat, Wali Mohammad Abdullah, Muhammad Nazrul Islam, “Towards Developing an Intelligent System to Suggest Optimal Path Based on Historic and Real-time Traffic Data,” 2017 20th International Conference of Computer and Information Technology (ICCIT), 22-24 December, 2017.
[17] YuanfangChenyx, Lei Shuyz), Lei Wang, “Poster Abstract: Traffic Flow Prediction with Big Data: A Deep Learning based Time Series Model,” 2017 IEEE Conference on Computer Communications Poster and Demo (INFOCOM'17 Poster/Demo).
[18] YuhanJia, Jianping Wu, and Yiman Du, “Traffic speed prediction using deep learning method,” 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).
[19] RongqiangQian, Yong Yue, FransCoeneny and Bailing Zhang, “Traffic Sign Recognition with Convolutional Neural Network Based on Max Pooling Positions,” 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).
[20] AbdelhadiAzzouni and Guy Pujolle, “A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction,” arXiv:1705.05690v3 cs.NI. 8 Jun 2017.
[21] Hao-Fan Yang, Tharam S. Dillon, Life Fellow, IEEE, and Yi-Ping Phoebe Chen, and Yi-Ping Phoebe Chen, “Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach,” IEEE transactions on neural networks and learning systems, , vol. 28, no. 10, october 2017.
[22] TianGuo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer,and Koichi Funaya, “Robust Online Time Series Prediction with Recurrent Neural Networks,” 2016 IEEE International Conference on Data Science and Advanced Analytics.