Traffic State Prediction using Convolutional Neural Network

Ratchanon Toncharoen
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand
fwk4work@gmail.com

Mongkut Piantanakulchai
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand
mongkut@siit.tu.ac.th

***Abstract*— Traffic state prediction methods have been considered by many researchers since accurate traffic prediction is an important part of the successful implementation of the Intelligent Transportation System (ITS). This study develops the traffic prediction model based on real traffic data in congested hours of expressways in Bangkok, Thailand. Unlike most studies, this model utilizes data from 40 nodes along the expressway instead of a single sensor. A Convolutional Neural Network (CNN) model was applied and compared to other widely used models. The result shows that the accuracy of CNN model is higher than other models.**

Keywords—Convolutional Neural Network; Traffic State Prediction; Classification; Intelligent Transportation System

I. INTRODUCTION

In many metropolitan areas, traffic congestion is a growing problem, and it will keep growing as there are more and more cars on the road. Congestion causes many effects to road users such as increasing travel time, wasting fuel, reducing their productivity [1], and increasing in carbon emission [2]. There are many solutions for traffic congestion like providing more roads or more public transport infrastructures, but these solutions may be capital intensive or time consuming. Thus, avoiding congested route by using Advance Traveler Information System (ATIS) is one of the alternative solutions.

Due to the benefits of accurate traffic prediction model, a lot of researches on traffic prediction have been conducted [3]. Statistical approaches have been widely used for traffic forecasting. The simplest statistical model like the historical average model is not a very effective approach for traffic prediction because it has no responses to unexpected events [4]. Hence, statistical time-series models which use mathematical model to fit past traffic behavior and then use it to forecast the future, are used to overcome this problem. An autoregressive integrated moving average model (ARIMA) is one of the most widely used statistical time series models [4, 5]. The modification of this model such as seasonal ARIMA model [6], ARIMA-GARCH model [7, 8], have been proposed by many studies to improve original ARIMA. Although these models often show notable results, they do not consider spatiotemporal effects, which are the important nature of traffic data. Alternative approaches for statistical models are nonparametric models. K-nearest neighbors (KNN), a nonparametric model that classifies data points based on theirs nearest data points, has been used by many studies dues to its simplicity of implementation, yet good performance [6, 9-14].

Another famous approach is the Artificial Neural Networks (ANN), which is a learning model inspired by human brain. ANNs have been used in many studies due to high predicting performance and capability of dealing with multi-dimensional data [15-20]. Despite their benefits of conventional ANN models, they are very prone to noise in data and unable to capture local dependency. Therefore, to deal with this disadvantage, a convolutional neural network (CNN), which is a type of advanced ANN, was proposed by Song et al. [21] for traffic speed prediction. Their proposed CNN model outperforms a multi-layer perceptron model (MLP), a type of feedforward neural network. The CNN model shows the ability to capture local dependencies and less vulnerable to noise in data. Another study applied CNN to predict traffic speed on large-scale transportation network [22]. Instead of using the data from single node as training data, they utilized the data along the network using GPS data recorded by a floating car, then converted them to space-time matrices, so that spatiotemporal information can be preserved. By using such data to train CNN, the performance of the model surpasses all other models including KNN, Random Forest (RF), and MLP. Despite many advantages of CNN model, very few studies utilized this model for traffic state prediction. Thus, it is very interesting to explore more on CNN model, and see how well this will performs on traffic state prediction using traffic data collected from many nodes along the expressway section.

In this paper, time-space matrices of traffic states are constructed using data the collected from 40 nodes along the expressway section in Bangkok, from Chaeng-Wattana entrance to Ngam-Wong-Wan exit. The traffic state data were collected every 10 minutes from free web-based resources. A CNN model is applied and compared to widely used model including KNN, RF, and MLP.

978–1–5386–5538–2/18/$31.00 ©2018 IEEE

II. Methodology

A. Data Collection and Preparation

Traffic data on Srirat expressway (Chaeng-Wattana Entrance to Ngam-Wong-Wan Exit) were collected every 10 minutes from a website that provided real time traffic information. This segment was chosen due to its recurrent traffic congestion in the morning on weekdays. The segment is divided into 40 nodes, with each node spanning 100 meters of distances. Node 1 is located on Chaeng-Wattana Entrance and Node 35 is located on Ngam-Wong-Wan Exit. Node 36-40 are 5 extra nodes planted for collecting the upstream traffic states. The website shows 3 colors of traffic state representing 3 degrees of congestion; green for free flow, yellow for moderate flow, and red for highly congested. Traffic states collected are converted to 40 by 30 time-space matrix, representing 40 nodes along the expressway section and 30 time-steps from 05:10 – 10:00 (1 time-step = 10 minutes). Colors of traffic state are considered as 3 discrete values; 1 for green, 2 for yellow, and 3 for red. Fig. 1 shows the visualization of converted data on 23/05/2016 from 05:10 to 10:00.

Fig. 1. Visualization of Observed Traffic States on 23/05/2016 (Monday)

The input data for the model are extracted to 5x5 matrices. Each matrix is used to predict future 10 minutes traffic state at focused location $V_{n,\ t+\Delta t}$. Each element in the matrix can be described as: $V_{n\pm k,\ t-i\Delta t}$ where $V_{n,\ t}$ represents the traffic state of Node ID (n) at time (t), Δt=10mins, $i\in\{0,1,2,3,4\}$, and $k\in\{0,1,2\}$. Table I shows the input matrix for predicting $V_{n,\ t+\Delta t}$, and Fig. 2 shows an example of visualization of input data used for predicting traffic state at 07:40 at node 13.

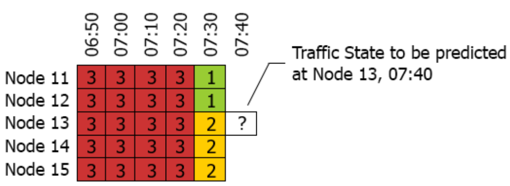

Fig. 2. Example of an Input Matrix for Predicting Traffic State

TABLE I. Input Matrix

Node ID	Time				
	t - 40	***t - 30***	***t - 20***	***t - 10***	***t***
n - 2	$V_{n-2,\ t-40}$	$V_{n-2,\ t-30}$	$V_{n-2,\ t-20}$	$V_{n-2,\ t-10}$	$V_{n-2,\ t}$
n - 1	$V_{n-1,\ t-40}$	$V_{n-1,\ t-30}$	$V_{n-1,\ t-20}$	$V_{n-1,\ t-10}$	$V_{n-1,\ t}$
n	$V_{n,\ t-40}$	$V_{n,\ t-30}$	$V_{n,\ t-20}$	$V_{n,\ t-10}$	$V_{n,\ t}$
n + 1	$V_{n+1,\ t-40}$	$V_{n+1,\ t-30}$	$V_{n+1,\ t-20}$	$V_{n+1,\ t-10}$	$V_{n+1,\ t}$
n + 2	$V_{n+2,\ t-40}$	$V_{n+2,\ t-30}$	$V_{n+2,\ t-20}$	$V_{n+2,\ t-10}$	$V_{n+2,\ t}$

By removing the days without congestion, traffic data of 132 days from 01/03/2016 to 15/08/2016 are used in the analysis presented in this study: 120 of them are used for training the model, and 12 of them are used for testing the performance. The data chosen for testing cover a variety of traffic state cases, including: Monday to Friday normal traffic state pattern, a day with extreme traffic congestion, and Saturday with traffic incident.

B. Convolutional Neural Network

CNN is a powerful learning model inspired by the visual cortex in the brain. CNN has been widely used for many applications due to its high performance on image and pattern recognition tasks [23]. The ability to extract important local features from input data makes CNN very suitable for traffic state prediction since the traffic state depends a lot on nearby values of traffic state in both time and space aspect.

This paper used a CNN model consisting of a convolutional layer, with a max pooling layer, followed by a fully connected layer with 20% dropout, and then an output layer. Fig. 3 shows overall architecture of the proposed model. The model was developed using the Keras toolkit with Tensorflow™ back end [24].

1) Convolutional Layer: The role of this layer is to extracts different features of the input by sliding each feature detector (kernel) through input matrix. The number of extracted features depends on the number of kernel used. For this study, 16 different kernels with the size of 2 by 2 are used. The output from convolutional layer will pass through activation function to introduce nonlinearity into the model. Retifier Linear Units (ReLUs) is chosen as the activation function since it does not squeeze the input, and increase the speed of training [23]. The ReLUs function can be written as (1).

$$f(x) = max(0,x) \qquad (1)$$

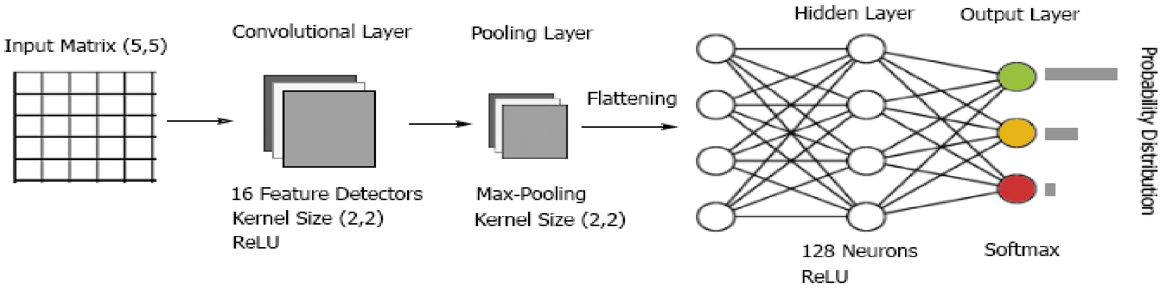

Fig. 3. Architecture of Proposed Convolutional Neural Network Model

2) Max-Pooling Layer: Pooling layer reduces the number of the parameters to train CNN, and makes the features robust to distortion and noise in data. Although there are many methods for pooling, max-pooling was widely used by many studies by reason of its performance to capture invariances in data compared to other methods [25, 26]. Max-pooling slides through matrices obtained from convolutional layer, and select the higest value in each section of the matrix, and construct a smaller matrix based on those values. Based on the recommendation from previous study by Ma et al. [22], this study uses max-poolings of size 2 by 2 for the pooling layer. The equation for max-pooling 2 by 2 can be described in (2) where x_{ij} represents the value of the input matrix at position i and j, and $y_{pooling}$ is the output.

$$y_{pooling} = max(x_{ij}),\ i \in \{1,2\},\ j \in \{1,2\} \quad (2)$$

3) Fully-Connected Layer: After all the outputs from pooling layers are concatenated into a vector (flattening), all members in that vector become the input nodes of a fully-connected layer. Like MLP, all the input nodes will be multiplied by a specific weight in each specific link. The summation of incoming values in each node in the hidden layer will be activated by activation function, then send out to output layer. This paper uses 128 neurons in a hidden layer, and ReLU as an activation function. Also, in order to reduce overfitting problem, a dropout of 20% is applied [27]. Noted that dropout is a technique of randomly setting some activations to 0 and force the model to find more way to classify image rather than relying too much on some features. This technique was proven to be useful by a successful image classification model, AlexNet [28].

4) Output Layer: Each nodes of this layer take outputs from every nodes in the hidden layer into an activation function, which will give the last output of the model. In our case, the data consists of 3 discrete variables, which can be considered as 3 labels categorical problem. Hence, the suitable function for the output layer is Softmax function. The output of softmax function shows a categorical distribution; in other words, a distribution of probabilities over different outcomes. The categorical cross entropy funtion is used for calculating loss to propagated back for weight adjustment in each link until the training is done.

C. Performance Evaluation

In this study, performances of each model are observed and compared based on the 2 following criteria.

1) Confusion Matrix: Since this problem is multilabel classification, the confusion matrix used to evalutate models. This matrix shows the numbers of correctly and incorrectly classified samples. Table II shows the confusion matrix used in this study. The accuracy of the model can be obtained from the confusion matrix using (3), where C_{ij} represents the occurances of traffic states that model predicts as j while the actual traffic state is i.

TABLE II. CONFUSION MATRIX

	Predicted			
	Traffic States	**1**	**2**	**3**
Actual	**1**	C_{11}	C_{12}	C_{13}
	2	C_{21}	C_{22}	C_{23}
	3	C_{31}	C_{32}	C_{32}

$$Accuracy = \frac{C_{11} + C_{22} + C_{33}}{\sum_{i,j} C_{ij}} \quad (3)$$

2) Root Mean Squared Error (RMSE): Although this problem is multilabel classification, it can also be viewed as 3 discrete integers. Thus, RMSE can also be used to evalutate the performance of the model. The RMSE gives some information that the accuracy from confusion can't. For example, if the actual traffic state is highly congested, by using confusion matrix accuracy, regardless of whether the model predicts out 1 (free flow) or 2 (moderate flow), the model gains no accuracy score. However, for RMSE, the model that predicts 2 will get lower error penalty than the model that predicts 1. RMSE can be written as (4).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\bar{y}_i - y_i)^2} \quad (4)$$

III. Results

Three widely used models are selected for comparison with the proposed CNN. The following list shows those 3 models and its setting. All models are trained and tested with the same data. Noted that the parameters of following models give the best performance on the dataset in this study comparing to other settings. In case of equal in performance, the setting with lowest computation cost will be selected.

- K-Nearest Neighbor (KNN), which is set to use 5 nearest points. (Chosen from 3, 5, 7, 9 nearest points.)
- Random Forest (RF), which is configured to utilized 10 trees. (Chosen from 10, 100 trees.)
- Multi-layer Perceptron (MLP), which contains 25 hidden nodes. (Chosen from 10, 25, 50, 100 ,128 nodes.)

Considering accuracy as criteria, from Fig. 4, the result shows that CNN outperforms other models with an accuracy of 0.89. The accuracy of MLP, KNN, and RF resemble at 0.87. Considering the comparison of confusion matrix between each model (Table III and IV), while MLP barely predicts moderate flow which are minorities in dataset, CNN correctly predicts moderate flow almost 5 times better than MLP. Fig. 5 shows the comparison of visualization between, observed traffic data, CNN prediction. It shows that CNN predicts traffic pattern closer than the observed pattern than MLP.

Table III. Confusion Matrix of CNN Model

		Predicted		
	Traffic States	**1**	**2**	**3**
Actual	**1**	4728	27	177
	2	341	66	260
	3	319	105	4777

Table IV. Confusion Matrix of MLP Model

		Predicted		
	Traffic States	**1**	**2**	**3**
Actual	**1**	4540	29	363
	2	367	14	286
	3	31	27	4863

For the RMSE, CNN outperforms other models with the lowest RMSE of 0.50. The MLP still gets a second place with RMSE of 0.56, followed by KNN with RMSE 0.58, and RF with RMSE with 0.59. Fig. 6 shows the comparison of RMSE.

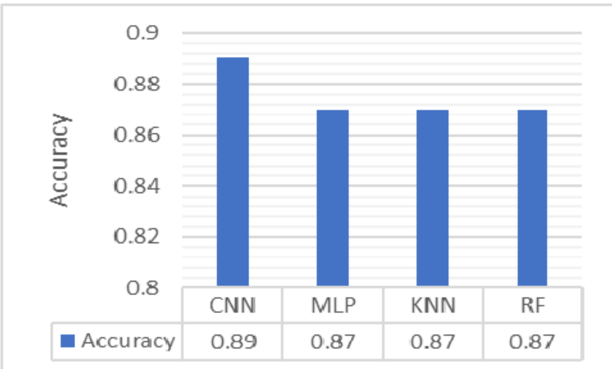

Fig. 4. Accuracy Comparison

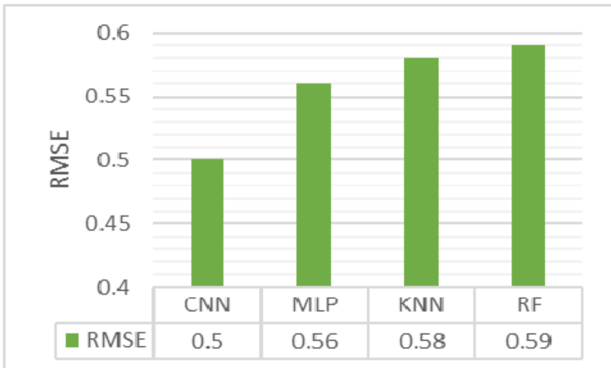

Fig. 6. Root Mean Squared Error Comparison

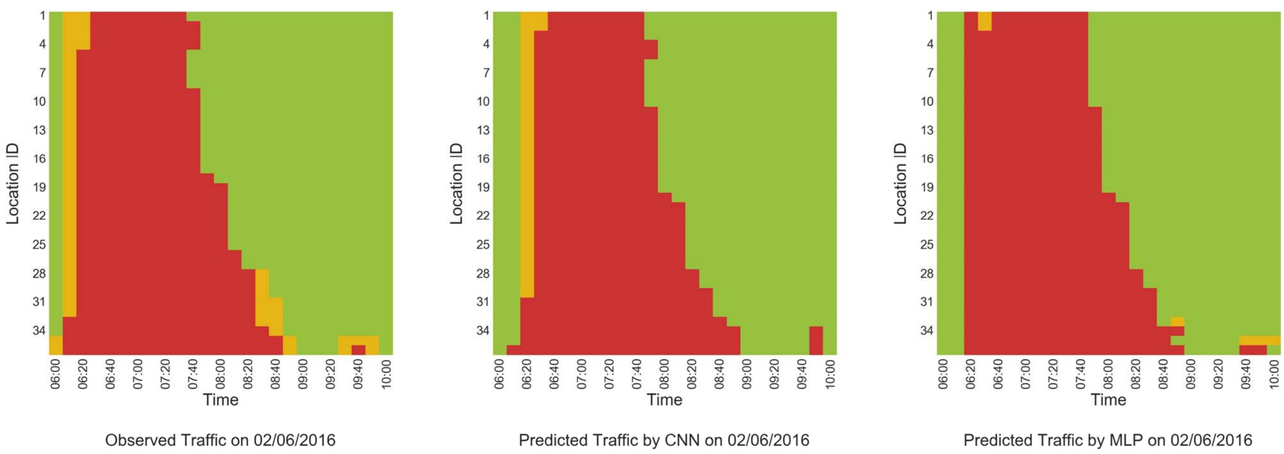

Fig. 5. Comparison of traffic states observed and predicted on 02/06/2016 (Thursday)

For validation, all models are assigned to predict traffic states on different route and time, which the model hasn't been trained with. The testing data are listed in Table V. The comparisons of accuracy and RMSE are shown in Fig. 7 and Fig. 8 respectively. The result shows that CNN still outperforms other models in this special task. Also, the model is proven to be robust since the accuracy drops down only by little when performs on different dataset.

TABLE V. LIST OF DATA FOR VALIDATION

Date	Time	Road Section
14/03/2016	16:20 – 20:20	Chalong Rat Expressway : Sukhumvit Entrace – Lat Phrao Exit
14/03/2016	16:00 – 20:00	Motor way 9 : Klong Luang Entrance – Lumlukka Exit
29/04/2016	19:50 – 23:50	Motor way 9 : Lumlukka Road Entrance – Rangsit Nakornnayok Exit
04/05/2016	19:40 – 23:40	Motor way 9 : Lumlukka Road Entrance – Rangsit Nakornnayok Exit
10/08/2016	16:20 – 20:20	Chalerm Mahanakorn Expressway : Ram Intra Entrance – Petch Buri Exit

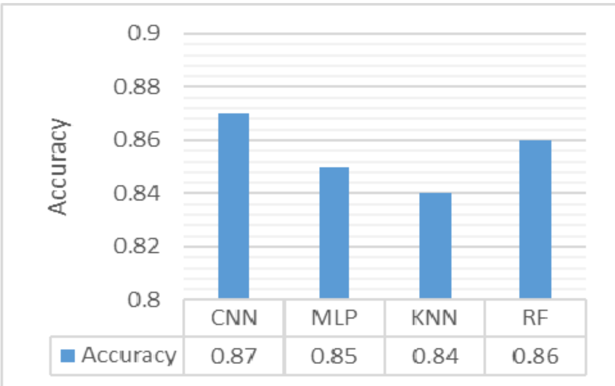

Fig. 7. Accuracy Comparison on Model's Validation

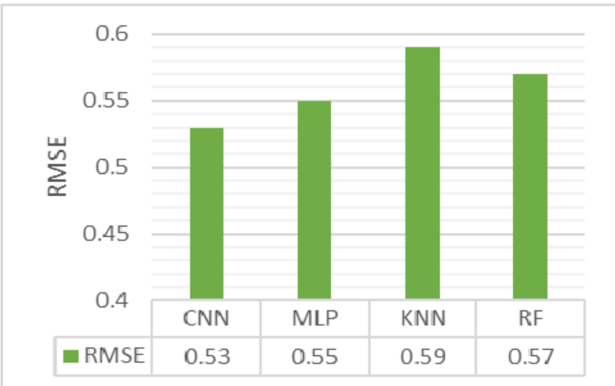

Fig. 8. Root Mean Squared Error Comparison on Model's Validation

IV. CONCLUSION

This paper proposes the CNN-based traffic state prediction due to its unique ability to capture local dependencies by extracting important local features from input. Instead of utilizing a single node like most studies, data from 40 nodes along the expressway segment are converted to space-time matrices and utilized to train CNN along with 3 models including KNN, RF, and MLP. The result shows that CNN outperforms other models with RMSE of 0.50. Even when assigned to predict traffic state on different route and time, CNN still performs well. However, the performance difference between CNN and MLP is not very significant. The possible reason is the input data size is not large enough for CNN to gain much advantage.

Finally, further improvements of the proposed model are suggested. For example, enlarging the input matrix until the accuracy won't increase, modification such as changing pooling function or activation function and compare how the performance will change. Increasing dimension of input matrix into 3D would be very interesting attempt. Adding more features than traffic states, such as, weekdays, binary variables telling if it's holiday or not, integers indicating degrees of rain at that moment, etc.

ACKNOWLEDGEMENT

The first author would like to thank Sirindhorn International Institute of Technology (SIIT), Thammasat University for providing him scholarship and facilities to make this research possible.

REFERENCES

[1] D. T. Hartgen, M. G. Fields, and A. T. Moore, "Gridlock and growth: the effect of traffic congestion on regional economic performance," Reason Foundation Los Angeles, CA2009.

[2] M. Barth and K. Boriboonsomsin, "Real-World Carbon Dioxide Impacts of Traffic Congestion," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2058, pp. 163-171, 2008/12/01 2008.

[3] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies,* vol. 19, pp. 387-399, 6// 2011.

[4] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of transportation engineering,* vol. 123, pp. 261-266, 1997.

[5] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-Term Prediction of Traffic Volume in Urban Arterials," *Journal of Transportation Engineering,* vol. 121, pp. 249-254, 1995.

[6] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies,* vol. 10, pp. 303-321, 8// 2002.

[7] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, pp. 607-612.

[8] Y. Zhang, Y. Zhang, and A. Haghani, "A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model," *Transportation Research Part C: Emerging Technologies,* vol. 43, Part 1, pp. 65-78, 6// 2014.

[9] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies,* vol. 62, pp. 21-34, 1// 2016.

[10] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies,* vol. 66, pp. 61-78, 5// 2016.

[11] Y. Kim, W. Kang, and M. Park, "Application of Traffic State Prediction Methods to Urban Expressway Network in the City of Seoul," *Journal of the Eastern Asia Society for Transportation Studies,* vol. 11, pp. 1885-1898, 2015.

[12] Y. Yin and P. Shang, "Forecasting traffic time series with multivariate predicting method," *Applied Mathematics and Computation,* vol. 291, pp. 266-278, 12/1/ 2016.

[13] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social and Behavioral Sciences,* vol. 96, pp. 653-662, 2013.

[14] Z. Zheng and D. Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies,* vol. 43, Part 1, pp. 143-157, 6// 2014.

[15] C. Goves, R. North, R. Johnston, and G. Fletcher, "Short Term Traffic Prediction on the UK Motorway Network Using Neural Networks," *Transportation Research Procedia,* vol. 13, pp. 184-195, 2016/01/01/ 2016.

[16] W. Hu, Y. Liu, L. Li, and S. Xin, "The short-term traffic flow prediction based on neural network," in *2010 2nd International Conference on Future Computer and Communication*, 2010, pp. V1-293-V1-296.

[17] K. Kumar, M. Parida, and V. K. Katiyar, "Short Term Traffic Flow Prediction for a Non Urban Highway Using Artificial Neural Network," *Procedia - Social and Behavioral Sciences,* vol. 104, pp. 755-764, 2013/12/02/ 2013.

[18] R. More, A. Mugal, S. Rajgure, R. B. Adhao, and V. K. Pachghare, "Road traffic prediction and congestion control using Artificial Neural Networks," in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, 2016, pp. 52-57.

[19] E. S. Yu and C. Y. R. Chen, "Traffic prediction using neural networks," in *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., IEEE*, 1993, pp. 991-995 vol.2.

[20] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transportation Research Part C: Emerging Technologies,* vol. 13, pp. 211-234, 6// 2005.

[21] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1293-1298.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, *Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction* vol. 17, 2017.

[23] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai*, et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition,* vol. 77, pp. 354-377, 2018/05/01/ 2018.

[24] Mart, #237, n. Abadi, P. Barham, J. Chen, Z. Chen*, et al.*, "TensorFlow: a system for large-scale machine learning," presented at the Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2016.

[25] Y. L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: Multi-way local pooling for image recognition," in *2011 International Conference on Computer Vision*, 2011, pp. 2651-2658.

[26] D. Scherer, A. Müller, and S. Behnke, *Evaluation of pooling operations in convolutional architectures for object recognition*, 2010.

[27] S. Gilda, "Source code classification using Neural Networks," in *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2017, pp. 1-6.

[28] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., ed: Curran Associates, Inc., 2012, pp. 1097-1105.