DUY TRAN QUANG, Ph.D.[1]
E-mail: tran1986@pukyong.ac.kr
SANG HOON BAE, Professor[2]
(Corresponding author)
E-mail: sbae@pknu.ac.kr
[1] Nha Trang University
Faculty of Civil Engineering
Nha Trang 57000, Vietnam
[2] Pukyong National University
Department of Spatial Information Engineering
Busan 48513, Korea

Intelligent Transport Systems (ITS)
Original Scientific Paper
Submitted: 13 July 2020
Accepted: 10 Nov. 2020

A HYBRID DEEP CONVOLUTIONAL NEURAL NETWORK APPROACH FOR PREDICTING THE TRAFFIC CONGESTION INDEX

ABSTRACT

Traffic congestion is one of the most important issues in large cities, and the overall travel speed is an important factor that reflects the traffic status on road networks. This study proposes a hybrid deep convolutional neural network (CNN) method that uses gradient descent optimization algorithms and pooling operations for predicting the short-term traffic congestion index in urban networks based on probe vehicles. First, the input data are collected by the probe vehicles to calculate the traffic congestion index (output label). Then, a CNN that uses gradient descent optimization algorithms and pooling operations is applied to enhance its performance. Finally, the proposed model is chosen on the basis of the R-squared (R^2) and root mean square error (RMSE) values. In the best-case scenario, the proposed model achieved an R^2 value of 98.7%. In addition, the experiments showed that the proposed model significantly outperforms other algorithms, namely the ordinary least squares (OLS), k-nearest neighbors (KNN), random forest (RF), recurrent neural network (RNN), artificial neural network (ANN), and convolutional long short-term memory (ConvLSTM), in predicting traffic congestion index. Furthermore, using the proposed method, the time-series changes in the traffic congestion status can be reliably visualized for the entire urban network.

KEYWORDS

*traffic congestion prediction; deep learning;
convolutional neural network; probe vehicles;
gradient descent optimization.*

1. INTRODUCTION

Traffic congestion leads to considerable wasted time and slow traffic, and it is one of the main challenges for traffic management agencies and traffic participants. To mention some examples of its detrimental effects on the economy, the traffic congestion costs in Seoul reached approximately 105,542 billion Korean Won (KRW) in 2013 [1], and the economic losses resulting from the traffic congestion in urban networks in the United States were estimated at $305 billion in 2017 [2]. To solve the congestion problem, a domain intelligent transport system (ITS) has been developed to enhance smoother, smarter, and safer journeys for traffic participants by adopting advanced techniques. The recent ITS applications have focused on the traffic system to find a solution for traffic problems, such as highway accident prediction [3], traffic speed prediction [4], and congestion prediction [5, 6].

In addition, the traffic congestion index does not have a specific standard. For instance, the Highway Capacity Manual 2010 (HCM2010) introduced the six-grade level of service (LOS) in the United States [7]. Xing et al. [8] established [0,100] as the range for a traffic congestion index based on the road characteristics and long-term observation. In addition, the local road network and congestion delay time were considered to estimate the traffic congestion index [9]. Feifei et al. [10] estimated the traffic congestion of urban road networks using the non-congestion state and average road speed. Nguyen et al. [11] evaluated an efficient traffic congestion index based on the density performance and velocity performance indices. Nevertheless, it is still difficult to collect comprehensive data, including travel speed, traffic density, and traffic volume. The Global Positioning System (GPS) has become an effective alternative tool for obtaining real-time traffic datasets [12]. The travel speed is used as an index for predicting traffic congestion.

Prediction of short-term traffic congestion has been a critical issue for ITS applications because more accurate information is required for traffic management agencies and stakeholders. Both short-term and long-term congestion predictions are based on real-time data to estimate the traffic congestion status in a time future. Furthermore, they depend on the data update rate for a few minutes to a few hours. In this work, the data update rate in a 5-minute interval is considered as short-term congestion prediction. In general, the methodologies applied for short-term traffic prediction are classified into two types: statistical methods and neural-network methods. The statistical methods are widely applied to predict traffic congestion. For example, Dawei et al. [13] applied OLS to a high-speed network, and Liu et al. [14] used a Bayesian network to predict urban road congestion based on road construction and bus development data. Lee et al. [15] applied a multiple linear regression model to the fusion of traffic congestion and weather factors. In addition, Liu and Wu [16] used the RF method to predict traffic congestion statuses. In conclusion, it was found that statistical algorithms cannot consider the spatiotemporal relationship of the traffic information and cannot be applied to the dynamic environment of road networks. As an improvement, artificial neural networks can work well for multi-dimensional and flexible data, and they have been applied to predict traffic congestion. For example, Mondal and Rehena [17] applied an ANN to the traffic congestion classification. The fusion of the ANN and decision tree has also been used for predicting the traffic congestion state on the basis of GPS data [18]. Nonetheless, the spatiotemporal features of traffic information could not be explained well.

Building upon these works, deep-learning algorithms have become potent approaches for traffic congestion prediction [19]. For instance, Zhang et al. [20] used a deep autoencoder neural network to forecast short-term traffic congestion in transportation networks. In addition, the attention-based long short-term memory (LSTM) was used to forecast traffic congestion, and it achieved superior performance over the autoregressive integrated moving average, KNN, and extreme gradient boosting methods [21]. Recently, based on internal memory units, the RNN managed to process temporal sequences by considering nonlinear traffic information and time-series datasets [22]. To deal with the vanishing gradient problems of long-term dependencies [23], many advanced RNNs have been proposed, including the gated recurrent unit (GRU) [24] and the stacked LSTM [25, 26]. For example, Sun et al. [27] showed that the LSTM and GRU models performed better for long-term traffic congestion prediction; it was also found that the stacked LSTM model is related to the temporal and spatial correlations of traffic speed [28]. Moreover, the CNN enabled the extraction of spatial features, and it has been applied for image recognition, object detection, and segmentation [29, 30]. Kurniawan et al. [31] applied the CNN for traffic congestion detection using CCTV camera images. Ma et al. [32] used the CNN for large-scale traffic speed prediction, and it achieved a great performance in comparison with the OLS, KNN, ANN, and RF methods. However, speed does not directly reflect the state of traffic congestion that could be measured by speed, travel time, delay, LOS, and congestion indices. In addition, they did not consider the effect of gradient descent optimization and pooling operations on prediction accuracy. The complexity of deep-learning models also results in the problem of overfitting. The short-term traffic state prediction performance was found to be better than the gradient descent optimization [33]. More importantly, the speed relationship has a large variance. Thus, the CNN model has the potential for short-term traffic congestion as it enables the extraction of the spatial relationship of the speed GPS data to guarantee the most important features.

In this study, we propose a hybrid deep CNN method that uses gradient descent optimization algorithms and pooling operations for predicting the short-term traffic congestion index using GPS probe vehicles. First, the input data are collected by GPS probe vehicles and geographic information systems (QGIS) from the Gangnam urban network in Seoul; the data comprise travel speed and free-flow speed (FFS), which are used for calculating and classifying the traffic congestion index according to HCM2010. Second, a hybrid CNN method that uses pooling operations (e.g. maximum and average pooling) and gradient descent optimization algorithms (e.g. stochastic gradient descent (SGD), adaptive learning rate (AdaDelta), and adaptive moment estimation (ADAM)), is applied to enhance the prediction performance of the CNN model. Finally, the R^2 and

RMSE are applied to choose a suitable approach for predicting the traffic congestion index. Overall, the major contributions of this work are as follows.

- An enhanced hybrid deep CNN is proposed that considers some gradient descent optimization algorithms (i.e., SGD, AdaDelta, and ADAM) and pooling operations (i.e., maximum and average pooling operations), for predicting the short-term traffic congestion index in an urban network using GPS probe vehicles. The traffic congestion index reflects the congestion situation in a time future. The experiment shows that the prediction accuracy is a significant improvement over the OLS, KNN, RF, ANN, RNN, and ConvLSTM methods.
- A set of hyperparameters is also proposed to explore the effect of important features on the traffic congestion prediction and to obtain reliable prediction performance at an urban network.
- The experimental results for a road network in the Gangnam area demonstrate the significant effect of the proposed method, and the time-series changes in the traffic congestion situation could be reliably predicted.

This paper is structured as follows. Section 2 presents the proposed model architecture, the traffic congestion index, the CNN, and the gradient descent optimization algorithms (OLS, KNN, RF, ANN, RNN, and ConvLSTM). Section 3 describes the conducted experiments and their results, and Section 4 concludes the study.

2. METHODOLOGY

2.1 Proposed model architecture

In this study, we proposed an approach to predict the short-term traffic congestion index based on the time-series sequence of the speed dataset. This approach is designed to improve important feature extraction and updated parameter optimization for predicting the traffic congestion index in urban road networks. In addition, the speed relationship has a large variance. Hence, the 5-minute intervals can express well the speed relationship and enhance the accuracy of traffic congestion prediction. The HCM2010 is applied to classify the traffic congestion situation. First, the raw data are transferred into the time sequence data through data pre-processing. Second, the time-series sequence data are converted to a time-space image (the input of the CNN model). Third, the traffic congestion index (the output label) is estimated by the running time and delay time. Fourth, the maximum pooling and average pooling are applied to improve important feature extraction. Furthermore, gradient descent optimization algorithms are used to enhance the updated parameter optimization. Next, the proposed model is compared with other models (OLS, KNN, RF, ANN, RNN, and ConvLSTM) in terms of R^2 and RMSE values. Finally, the time-series changes in the traffic congestion status in urban networks are predicted at 5-minute intervals. We also discuss the proposed model in more detail in the following sections. *Figure 1* shows the research flow architecture.

2.2 Traffic congestion index

The travel speed is an important service measure for the state of an urban network. According to the HCM2010, the LOS on an urban road depends on the average travel speed, which is measured using the running time and the delay time. The traffic congestion index (TCI) is calculated as follows.

$$TCI = I_s I_{TTI} \quad (1)$$

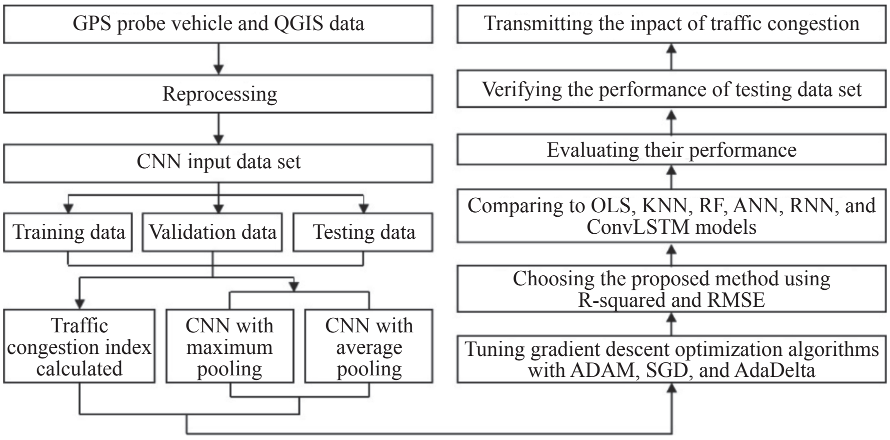

GPS probe vehicle and QGIS data
Reprocessing
CNN input data set
Training data
Validation data
Testing data
Traffic congestion index calculated
CNN with maximum pooling
CNN with average pooling
Tuning gradient descent optimization algorithms with ADAM, SGD, and AdaDelta
Choosing the proposed method using R-squared and RMSE
Comparing to OLS, KNN, RF, ANN, RNN, and ConvLSTM models
Evaluating their performance
Verifying the performance of testing data set
Transmitting the inpact of traffic congestion

Figure 1 – Research flow architecture

$$I_s = \frac{v_i}{v_{max}} \tag{2}$$

$$I_{TTI} = \frac{T_r}{T_t} \tag{3}$$

where I_s is the mean speed performance index, I_{TTI} is the time performance index, v_i is the observation speed of the i^{th} vehicle, v_{max} is the FFS of each link, and T_r and T_t denote the running time and the total observation time of each link, respectively.

The evaluation criterion for the traffic congestion index on the urban road based on the LOS classification from HCM2010 is shown in *Table 1*.

Table 1 – Traffic congestion index (Source: HCM2010)

TCI values	Congestion status
>0.9	Free flow
(0.7,0.9]	Reasonable flow
(0.5,0.7]	Near free flow
(0.4,0.5]	Mild congestion
(0.25,0.4]	Moderate congestion
≤0.25	Heavy congestion

2.3 Convolutional neural network

CNN, introduced by Hubel and Wiesel [34], is a special type of deep neural networks. The basic structure of a CNN in the context of the traffic congestion index comprises two convolutional hidden, two pooling, and fully connected layers. CNN has become a promising method for traffic congestion prediction thanks to extracting important features. Firstly, the CNN extracts spatial hierarchies of features automatically through convolutional hidden and pooling layers. Finally, the output of the feature extraction is mapped through a fully connected layer. *Figure 2* presents the CNN architecture in the context of the traffic congestion status, including the input data, feature extraction, prediction, and regression output.

After the feature extraction process, the filter features are flattened and fed into the fully connected layer. The output is then interpreted by the rectified linear unit (ReLU) function; it directly extracts the input if it is positive or else extracts zero. The main advantage of the ReLU function is a reduced likelihood of vanishing gradient. This benefit makes it faster learning and more beneficial than other functions. In particular, deep neural networks with the ReLU function have a better convergence performance than the tanh function [35]. Thus, the ReLU function is applied to the CNN model because of the ease of training and better performance. First, the time-series sequence data of the traffic speed is converted into a time-space image to generate the CNN input data, as shown here [36].

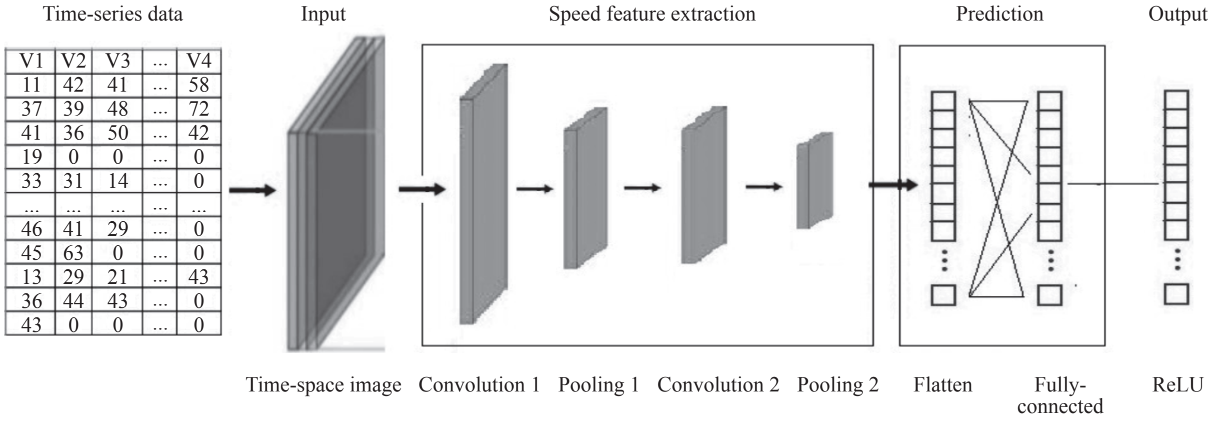
$$x^i = [m_i, m_{i+1}, \ldots, m_{i+P-1}], \;\; i \in [1, N-P-F+1] \tag{4}$$

where N is the length of the time intervals, F is the length of the input time intervals, P is the length of the output time intervals, i denotes the sample index, and m_i denotes the column vector

Second, the convolutional layer identifies the patterns between the time intervals. In other words, the convolution layer uses small filters to generate a feature map with a higher level of representation. The output feature map is convolved with the convolution kernel and processed by the activation function to generate the feature map for the following layer; the process can be expressed using the following equation.

$$x_j^l = \sigma\left(\sum_{i=1}^{l-1} x_i^{l-1} \cdot k_{ij}^l + b_j^l\right) \tag{5}$$

where σ is the ReLU activation function, x_j^l is the j^{th} feature map of the l^{th} layer, x_i^{l-1} is the i^{th} feature map of the $(l-1)^{th}$ layer, k_{ij}^l is the convolution kernel, and b_j^l is the bias term.



V1	V2	V3	...	V4
11	42	41	...	58
37	39	48	...	72
41	36	50	...	42
19	0	0	...	0
33	31	14	...	0
...	...	...	...	...
46	41	29	...	0
45	63	0	...	0
13	29	21	...	43
36	44	43	...	0
43	0	0	...	0

Figure 2 – CNN architecture in the context of the traffic congestion status

The pooling layer is applied to reduce the dimensionality and preserve the important information; the common categories of the pooling operations are average pooling and maximum pooling. Considering some important features, this study explores and chooses the appropriate pooling layer. The maximum pooling operation calculates the maximum value for each filter of the feature map, and the average pooling operation computes the average value for each filter of the feature map. The process can be expressed as follows.

$$y_{pool} = pool(d_{ef}), \ \ e \in [1 \ldots p], \ \ f \in [1 \ldots q] \tag{6}$$

where y_{pool} denotes the pooling output, d_{ef} is the data value of the input matrix at positions e and f, p and q are the two dimensions of the pooling size, and *pool* denotes the maximum or average pooling operations.

The output of the traffic speed feature extraction is converted to a dense vector using the flatten output layer, which is a higher-level feature of the time-series sequence of the traffic speed data. This output is then fed into the fully connected layer. The flatten output layer with L as the depth of the CNN model can be expressed as follows.

$$y^{L}_{flatten} = flatten([y^{L}_{1}, y^{L}_{2}, \ldots, y^{L}_{j}]) \tag{7}$$

Finally, the CNN training process comprises forward and backward propagation. The forward process predicts the output value (traffic congestion index) from the input feature matrix (traffic speed dataset), and the backward process updates the weight of the CNN model to minimize the cost function. The backward propagation is discussed in more detail in section 2.4.

2.4 Gradient descent optimization algorithms

Gradient descent optimization [35] is one of the best methods for training deep neural networks. Basically, this method is used for minimizing the objective function by updating the parameters in the opposite direction. In this study, we compare SGD, AdaDelta, and ADAM to determine the best optimization algorithm for traffic congestion prediction.

First, the SGD uses a batch size of only one per iteration. In the SGD, each batch is randomly chosen, and the SGD moves the redundant data by getting one update at a time [37]. Hence, the SGD is affected by noise.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta, x^{(i)}, y^{(i)}) \tag{8}$$

where $x^{(i)}$ denotes the input data, $y^{(i)}$ denotes the labels, η denotes the learning rate, $J(\theta)$ denotes the objective function, and ∇_{θ} denotes the objective function gradient.

Second, instead of considering all the past gradients, the AdaDelta is based on the moving window of the gradient updates [38], and it also uses a component to enhance the acceleration learning. Therefore, the AdaDelta improves the learning rate converging to zero as time progresses.

$$E[g^2]_t = \rho \cdot E[g^2]_{t-1} + (1 - \rho) g^2_t \tag{9}$$

where $E[g^2]_t$ denotes the expected value of the accumulate gradient at time t, $E[g^2]_{t-1}$ denotes the expected value of the accumulate gradient at time $t-1$, ρ is the decay constant, and g_t is the gradient of the parameters.

Third, ADAM optimizes the stochastic objective functions according to the estimation of the averages of both the gradients and the second moments of the gradients of the historical gradients [39]. Moreover, it is suitable for big data and requires low memory space.

$$E[v_t] = E[g^2_t] \cdot (1 - \beta^2_t) + \zeta \tag{10}$$

where $E[v_t]$ is the expected value of the exponential moving average at a timestep t, g_t is the gradient of the parameters, and $E[g^2_t]$ is the true second moment.

2.5 Performance comparison

To evaluate the superiority of the proposed model, the hybrid CNN model was compared with other models, namely the OLS, KNN, RF, ANN, RNN, and ConvLSTM models. The OLS regression model is the simplest statistical algorithm, and the KNN regression model applies the similarity of features to forecast new data points. The RF regression model uses many individual decision trees. In this work, the KNN regression model used a grid search to determine the optimal nearest points. The RF was set to produce 200 decision trees.

The ANN uses data-processing techniques that are based on how the human brain processes information. The multilayer perceptron (MLP), which was introduced by Rumelhart et al. [40], is a development over the original ANN. In this study, the MLP comprised multiple layers, including an input layer, a hidden layer, and an output layer. The ANN was configured to apply 100 hidden units in a hidden layer.

The RNN, initially developed by James and Rumelhart [41], is a special type of feed-forward model owing to its ability to remember the past for sequence processing. In this study, the stacked LSTM model was optimized to contain three hidden LSTM layers with 512, 512, and 512 hidden units in each layer. Furthermore, because the ConvLSTM [35] is like the LSTM model, the internal matrix multiplications were replaced by the convolution operations. In this work, the ConvLSTM model included two convolutional hidden layers, two average pooling layers, and one LSTM layer.

3. EXPERIMENTS

3.1 Data description

The Gangnam district has the highest congestion level in Seoul, and the urban network in Gangnam witnesses big traffic volumes. Hence, the Gangnam district was chosen as the study area, with 24 main links, as shown in *Figure 3*. In the field, a large number of input data must be obtained. However, in this study, we limited our focus on the traffic data to one month. The input data were collected using GPS probe vehicles for a period of one month (1 July 2015–31 July 2015). The research area has about 5500 probe vehicles per day. The total number of speed data is about 7.2 million. A GPS receiver was associated with a transportation management center to record the travel speed, travel time, segment lengths, and FFS of the probe vehicles at 5-minute intervals to account for the fluctuation over shorter time intervals. The raw data obtained using the GPS probe vehicles and the QGIS data are shown in *Tables 2 and 3*, respectively.

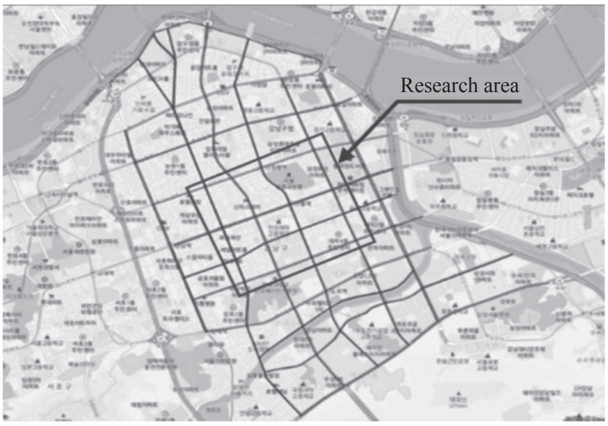


Figure 3 – Research area

Finally, the raw data obtained using the GPS probe vehicles and the QGIS data were integrated via data pre-processing. This process also separated the data into 5-minute intervals for each link to generate a time-series sequence of the input data, as shown in *Table 4*. We selected the first 16 days as the training set, 4 continuous days as the validation set, and the last 11 days as the testing set.

Table 2 – Raw data obtained using GPS probe vehicles

Vehicle ID	Date	Time	5-minute interval	Passed link ID	Heading link ID	Average speed [km/h]
6804301	20150701	1853	3	1130000501	1130000103	27
6804301	20150701	1910	3	1130000103	1130000102	22
…	…	…	…	…	…	…
6804301	20150701	2054	4	1130001400	1130001001	72

Table 3 – Raw data based on the QGIS

No.	Link ID	Number of lanes	Free-flow speed [km/h]	Distance [m]
1	1220035304	4	60	36.06068392
2	1220035204	4	60	36.38352808
…	…	…	…	…
24	1220036902	3	60	234.4012039

Table 4 – Time-series sequence of the input dataset

Time intervals [s]	TCI	V1	V2	V3	V4	V5	…
5	0.48	11	42	41	58	0	…
10	0.82	37	39	48	72	0	…
…	…	…	…	…	…	…	…
2400	0.24	19	9	34	34	36	…

3.2 Parameter settings and evaluation metrics

Hyperparameter optimization is an important role to build a perfect model architecture. However, there is no specific rule to choose the number of hidden layers and the number of neurons in each hidden layer. The trial-and-error method has become the most reliable approach to decide hyperparameters for a specific problem through systematic experimentation. In this work, the trial-and-error method was applied to configure the set of hyperparameters in the traffic congestion context according to RMSE and R^2. In addition, the cross-validation method was used to evaluate the accuracy of the testing dataset. The set of hyperparameters of the CNN model are shown in *Table 5*. Based on our experiment, the prediction accuracy performs well at the number of hidden layers of 2.

The prediction accuracy was evaluated using RMSE and R^2. The RMSE results in negatively oriented scores, which means that the model with a lower value is better. The RMSE measures the average squared values between the prediction and actual observation values, as shown below.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}i))^2} \qquad (11)$$

Table 5 – Set of hyperparameters of the hybrid CNN model

Layers	Name	Hyperparameters
Layer 1	Convolution	Filter = 128, Kernel size = 2
Layer 2	Pooling	Pooling size = 2
Layer 3	Convolution	Filter = 64, Kernel size = 2
Layer 4	Pooling	Pooling size = 2
Layer 5	Flatten	
Layer 6	Fully connected	Output shape = 64
Layer 7	Fully connected	Output shape = 32
Layer 8	Output	Output shape = 1

where y_i denotes the true value, $\hat{y}_i$ denotes the predicted value, and n denotes the number of observations.

R^2 measures the percentage of the variance for a dependent variable that is explained by independent variables. In addition, it calculates the strength of the relationship between the regression model and the dependent variables. R^2 ranges from 0 to 1, which means that the model with a higher value is better, as shown below.

$$R^2 = 1 - \frac{Unexplained\ Variation}{Total\ Variation} = 1 - \frac{SS_{regression}}{SS_{Total}} \qquad (12)$$

where $SS_{regression}$ denotes the sum squared regression error, and SS_{Total} denotes the sum squared total error.

4. 4. RESULTS AND DISCUSSION

4.1 Improving accuracy

To enhance the accuracy of the CNN model, we considered the effects of the gradient descent optimizations (ADAM, SGD, and AdaDelta) and pooling operations (maximum and average pooling) for a hybrid CNN model. *Tables 6 and 7* and *Figure 4* show the results of the hybrid CNN model for the training and validation datasets. The CNN model with the average pooling and ADAM algorithm (proposed model) achieved the highest prediction performance with the training and validation average R^2 of 0.985 and 0.976, respectively, followed by the CNN model with the maximum pooling and the ADAM algorithm (0.982 and 0.974). This indicates that the proposed model explained 98.5% and 97.6% of the variance in the dependent variable for the training and validation datasets, respectively. Similarly, the proposed model achieved the lowest prediction error with the training and validation average RMSE values of 0.0345 and 0.0433, respectively, followed by the CNN model with the maximum pooling and ADAM algorithm (0.0367

Table 6 – Results of the hybrid CNN model for the training and validation datasets

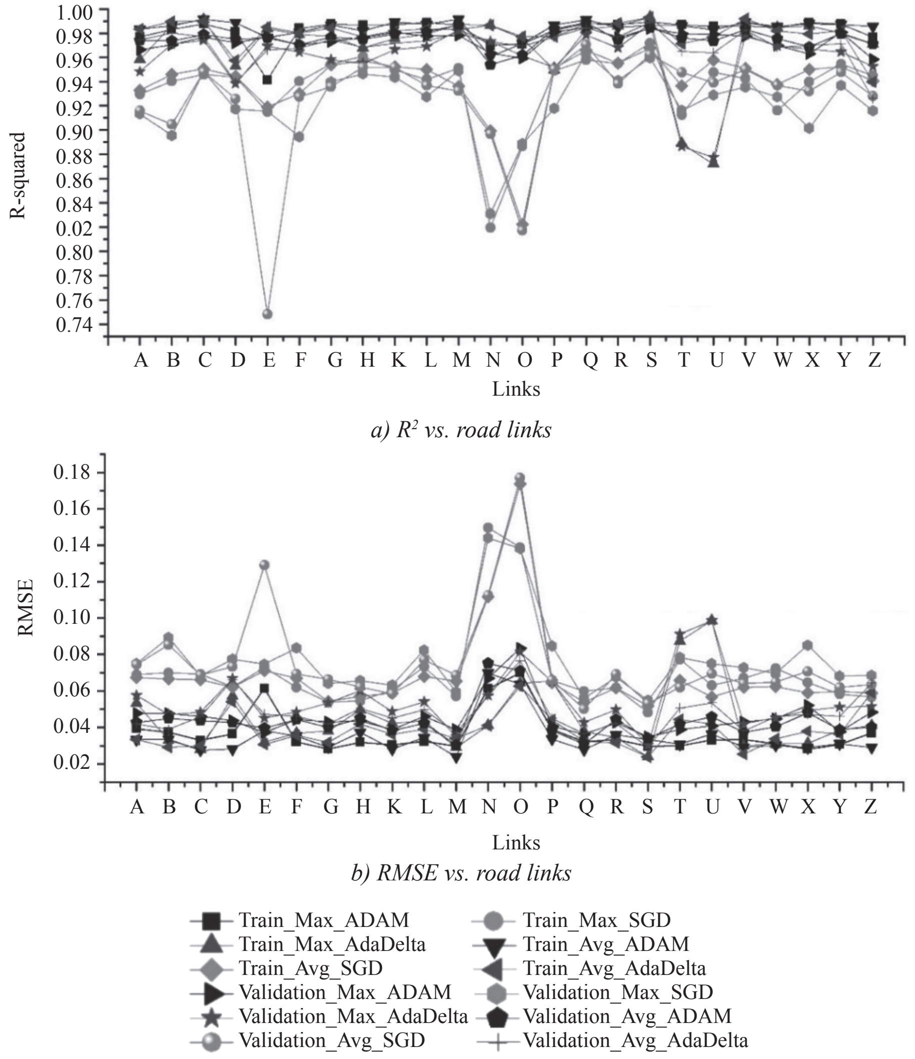
Metrics	Training datasets					
	Maximum pooling			Average pooling		
	ADAM	SGD	AdaDelta	ADAM	SGD	AdaDelta
Average R^2	0.982	0.936	0.970	0.985*	0.940	0.980
STD of average R^2	0.0102	0.0309	0.0300	0.0068*	0.0291	0.0114
Average RMSE	0.0367	0.0703	0.0445	0.0345*	0.0686	0.0379
STD of RMSE	0.0112	0.0239	0.0186	0.0106*	0.0251	0.0097

** indicates the best result*

Table 7 – Results of the hybrid CNN model for the training and validation datasets

Metrics	Validation datasets					
	Maximum pooling			Average pooling		
	ADAM	SGD	AdaDelta	ADAM	SGD	AdaDelta
Average R^2	0.974	0.923	0.959	0.976*	0.925	0.970
STD of average R^2	0.0084	0.0282	0.0263	0.0076*	0.0479	0.0133
Average RMSE	0.0453	0.0778	0.0550	0.0433*	0.0757	0.0476
STD of RMSE	0.0104	0.0216	0.0162	0.0102*	0.0275	0.0104

** indicates the best result*



a) R^2 vs. road links

b) RMSE vs. road links

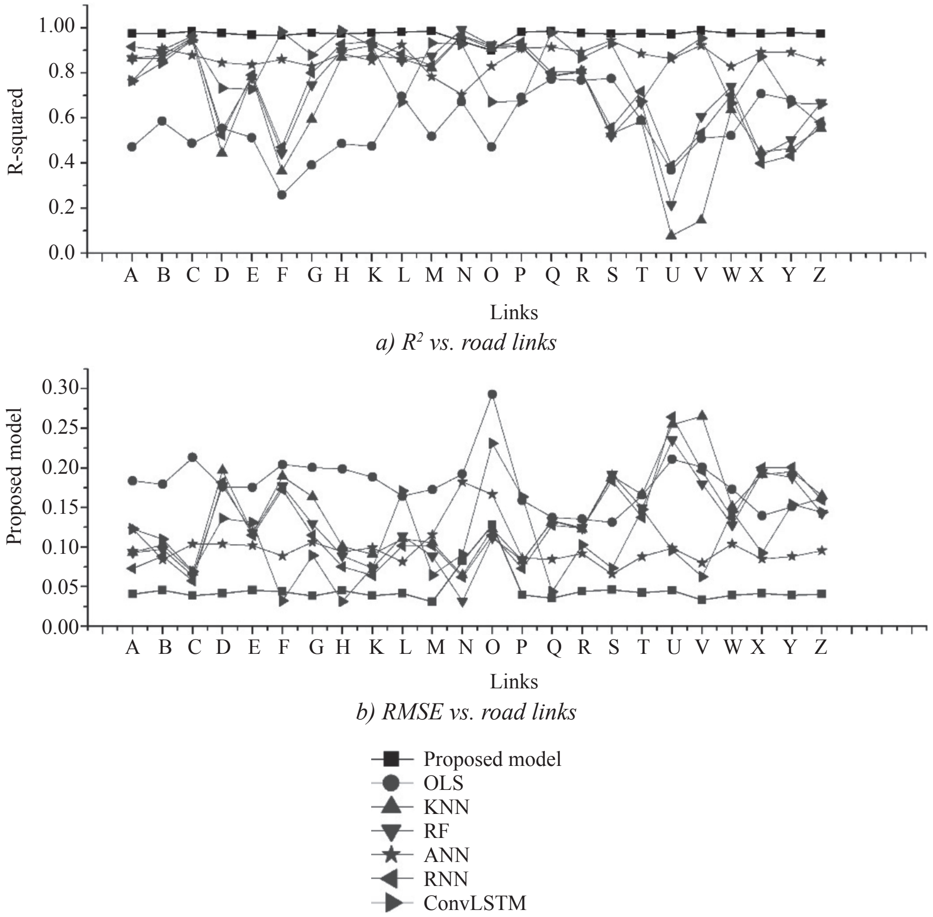
Figure 4 – Results of the hybrid CNN model

and 0.0453, respectively). *Figure 4* indicates that the fluctuation range of the proposed model was smaller than that of the others, corresponding to the lowest STD of the average R^2 and RMSE values of 0.0068 and 0.0106 (training datasets) and 0.0076 and 0.0102 (validation datasets), respectively. The experiments confirm that the proposed model operates more stably and efficiently than the other models in all circumstances studied. Therefore, the proposed model is a suitable method for predicting the traffic congestion index.

4.2 Performance evaluation

To verify the superiority of the proposed model, other algorithms were chosen for comparison, namely the OLS, KNN, RF, ANN, RNN, and

ConvLSTM algorithms. *Figure 5 and Table 8* show the results of the different models for the entire links in terms of R^2, RMSE, and STD. In all circumstances, the proposed model outperformed the other models in the testing datasets, implying that the proposed model can better predict the traffic congestion index. As seen in *Table 8*, the proposed model achieved the highest prediction performance with an average R^2 of 0.972, followed by the ANN (0.862), ConvLSTM (0.824), and RNN (0.737). This means that the proposed model explained 97.2% of the variance in the dependent variable. Similarly, the proposed model achieved the lowest prediction error with an average RMSE of 0.0460, followed by the ANN (0.1008), ConvLSTM (0.1075), and RF (0.1298). Furthermore, the fluctuation range of the proposed model was smaller than that of the other algorithms in terms of R^2 and RMSE, with STD values of 0.0178 and 0.0197, respectively. As shown in *Figure 5*, the proposed model achieved the highest R^2 and the lowest RMSE values among all the entire links. Therefore, the proposed model significantly outperforms the other algorithms under all the circumstances in predicting the traffic congestion index.



a) R^2 vs. road links

b) RMSE vs. road links

Figure 5 – Results of the different models

Table 8 – Results of the different models for the testing datasets

Models	Average R^2	STD of average R^2	RMSE	STD of RMSE
OLS	0.564	0.1347	0.1795	0.0342
KNN	0.671	0.2511	0.1435	0.0549
RF	0.730	0.2011	0.1298	0.0494
ANN	0.862	0.0554	0.1008	0.0259
RNN	0.737	0.1979	0.1303	0.0546
ConvLSTM	0.824	0.1231	0.1075	0.0488
Proposed model	0.972*	0.0178*	0.0460*	0.0197*

** indicates the best result*

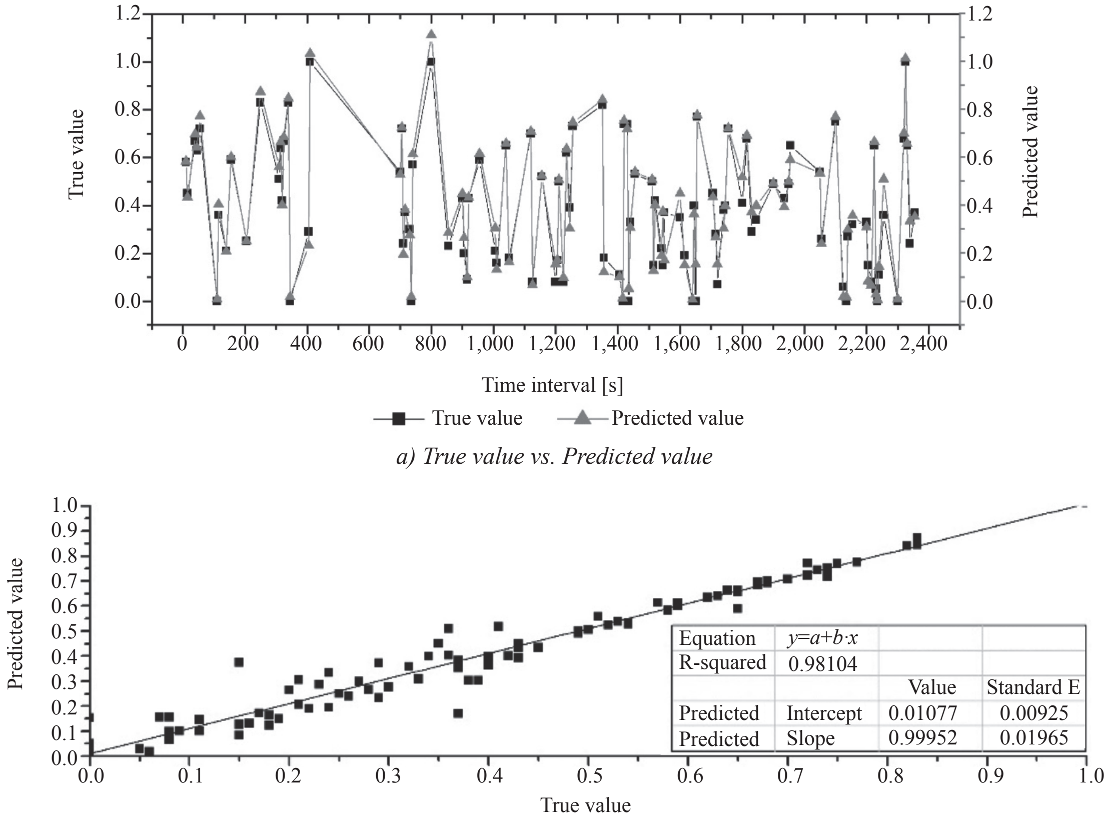
4.3 Testing dataset performance

The testing dataset is applied to confirm the performance of the proposed model. *Figure 6* shows a comparison between the true and predicted values for a specific link in one day. As seen in *Figure 6a*, the predicted values were close to the true values with a small RMSE of 0.0528. In *Figure 6b*, the scatter plot is almost linear with an R^2 of 0.98104. This means that the proposed model explained 98.104% of the variance in the predicted value. Therefore, the performance of the prediction model was a good fit for the learning algorithm for the testing datasets. Thus, the CNN model with the average pooling and ADAM algorithm is suitable for predicting the traffic congestion index.

4.4 Time-series changes in the traffic congestion index

Considering the urban network, *Figure 7* and *Table 9* show the average TCI values for the study area in one day at different time periods. A single day was divided into three time periods: morning peak hours (07:00–09:00), afternoon peak hours (12:00–14:00), and evening peak hours (18:00–20:00). The average TCI of the morning peak hours (0.41) was higher than that of the afternoon (0.30) and evening (0.24) peak hours. According to *Table 1*, the traffic congestion status during the morning peak hours (mild congestion) was less than that during the afternoon peak hours (moderate congestion) and evening peak hours (heavy congestion). Thus, the traffic congestion status during the morning peak hours was better than that of the afternoon and evening peak hours for the entire research area.

To visualize the time-series changes in the traffic congestion status for the entire research area at 5-minute intervals, *Figure 8* shows the relationships between the entire links, time, and the TCI, where the points are color-coded based on the TCI. The points closer to the top denote low traffic congestion, whereas those closer to the bottom denote heavy traffic congestion. During the morning peak hours, almost all the points were closer to the top than in the afternoon and evening peak hours. Therefore, the traffic congestion status during the morning peak hours is better than that during the afternoon and evening peak hours.



a) True value vs. Predicted value

b) Scatter plot between true value vs. predicted value

Figure 6 – Comparison between the true and predicted values for one specific link

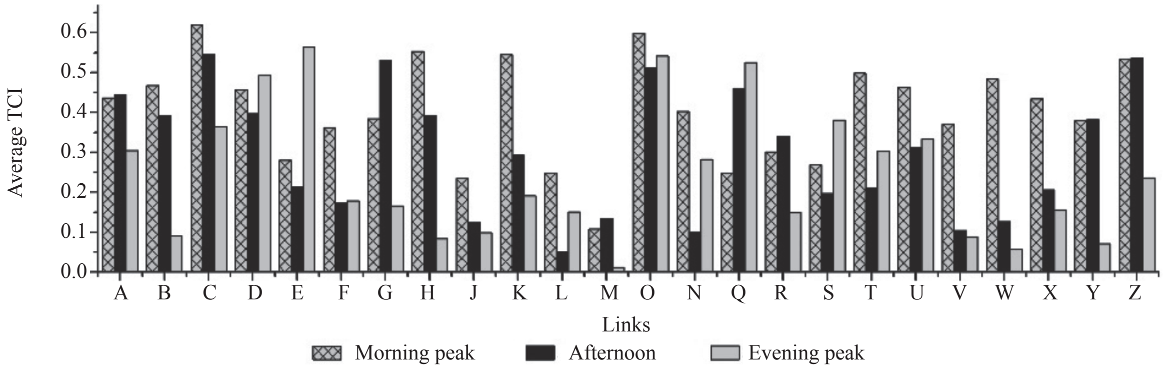


Figure 7 – Average TCI for the study area in one day at different time periods

Table 9 – Traffic congestion index vs. time periods

	Time periods		
	Morning peak hours	Afternoon peak hours	Evening peak hours
Average TCI	0.41	0.30	0.24

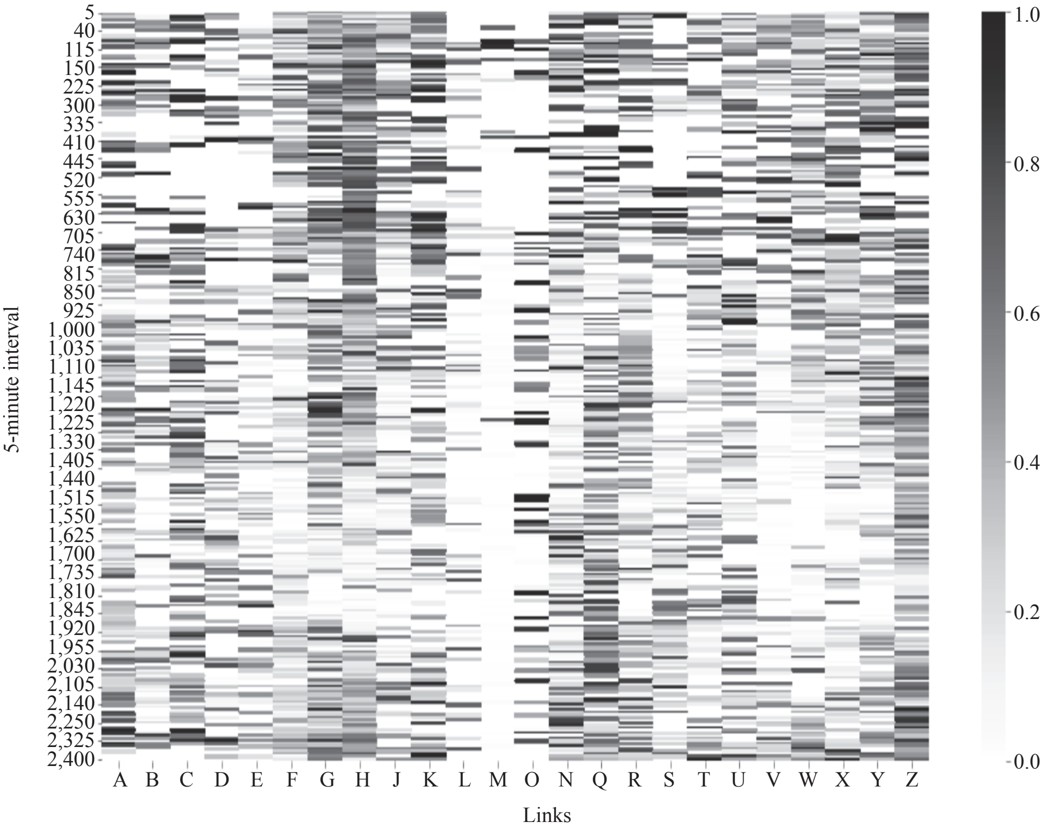


Figure 8 – Heat map of the traffic congestion index pattern for the entire links in one day

Consequently, the experimental results show that the hybrid CNN model that uses the average pooling and ADAM algorithm significantly improves the prediction performance of the traffic congestion index. In addition, the proposed model achieved the highest prediction accuracy and the lowest prediction error in comparison with the other models, such as the OLS, KNN, RF, ANN, RNN, and ConvLSTM models. In the best-case scenario, the R^2 value of the proposed model was 98.7%. Furthermore, based on the heat map of the time-series changes in the traffic congestion status, the traffic congestion status for the entire links during the morning peak hours was better than that during the afternoon and evening peak hours. In addition, unexpected events, such as accidents, lane blockage, can create new traffic congestion in the off-peak hours and extend the delay time for traffic congestion. In this study, we did not analyze the impact of unexpected events. However, a month of data included unexpected events as well. In our future work, we will consider the impact of unexpected events to guarantee traffic congestion accuracy.

5. CONCLUSION

In summary, the proposed hybrid CNN model that uses the average pooling and ADAM algorithm showed superior performance compared to other algorithms, namely the OLS, KNN, RF, ANN, RNN, and ConvLSTM algorithms, in terms of the R^2 and RMSE values. The main contributions of this work include two characteristics, namely (a) the proposed hybrid approach, which can predict with high accuracy the traffic congestion index in urban networks using GPS probe vehicles; and (b) the time-series changes in the traffic congestion status could be reliably predicted by adopting the hybrid CNN model that uses the average pooling and ADAM algorithm. Furthermore, this study presents positive effects in choosing the optimal route for vehicles. In future studies, we will consider comprehensive input data (e.g. weather, accident, lane blockage, traffic volume, and road geometric) and involve more realistic predictions by expanding the research area.

TRẦN QUANG DUY, Nghiên cứu sinh[1]
Địa chỉ mail: tran1986@pukyong.ac.kr
SANG HOON BAE, giáo sư[2]
(Đồng tác giả)
Địa chỉ mail: sbae@pknu.ac.kr
[1] Đại học Nha Trang, Khoa Xây dựng
Nha Trang 57000, Việt Nam
[2] Đại học Quốc gia Pukyong
Khoa Kỹ thuật Thông tin Không gian
Busan 48513, Hàn quốc

PHƯƠNG PHÁP MẠNG NƠ-RON TÍCH CHẬP SÂU ĐỂ DỰ ĐOÁN CHỈ SỐ TẮC NGHẼN GIAO THÔNG

TÓM LƯỢC

Ùn tắc giao thông là một trong những vấn đề quan trọng nhất ở các thành phố lớn, và tốc độ di chuyển tổng thể là một yếu tố quan trọng phản ánh tình trạng giao thông trên mạng lưới đường bộ. Nghiên cứu này đề xuất phương pháp mạng nơ-ron tích chập sâu (CNN) sử dụng các thuật toán tối ưu hóa độ dốc và các hoạt động tổng hợp để dự đoán chỉ số tắc nghẽn giao thông ngắn hạn trong các mạng đô thị dựa trên các phương tiện thăm dò. Đầu tiên, dữ liệu đầu vào được thu thập bởi các phương tiện thăm dò để tính toán chỉ số tắc nghẽn giao thông (nhãn đầu ra). Sau đó, CNN sử dụng các thuật toán tối ưu hóa độ dốc và các hoạt động tổng hợp được áp dụng để nâng cao hiệu suất của nó. Cuối cùng, mô hình đề xuất được chọn trên cơ sở các giá trị R bình phương (R^2) và sai số bình phương trung bình (RMSE). Trong trường hợp tốt nhất, mô hình đề xuất đạt được giá trị R^2 là 98,7%. Ngoài ra, các thử nghiệm cho thấy mô hình được đề xuất vượt trội hơn đáng kể so với các thuật toán khác, cụ thể là bình phương nhỏ nhất thông thường (OLS), k-láng giềng gần nhất (KNN), rừng ngẫu nhiên (RF), mạng nơ-ron hồi quy (RNN), mạng nơ-ron nhân tạo (ANN), và mạng bộ nhớ dài-ngắn tích chập (ConvLSTM), trong việc dự đoán chỉ số tắc nghẽn giao thông. Hơn nữa, bằng cách sử dụng phương pháp được đề xuất, những thay đổi theo chuỗi thời gian của tình trạng tắc nghẽn giao thông có thể được thể hiện một cách đáng tin cậy cho toàn bộ mạng lưới đô thị.

TỪ KHÓA

Dự báo tắc nghẽn giao thông; học sâu; mạng nơ-ron tích chập; xe thăm dò; tối ưu hóa dốc.

REFERENCES

[1] Ko JH. *Seoul's Transportation Demand Management Policy*. Seoul, Korea: The Seoul Institute; 2015.

[2] Cookson G, Pishue B. *INRIX Global Traffic Scorecard*. Kirkland, Washington: INRIX research; 2018.

[3] Yasin Çodur M, Tortum A. An Artificial Neural Network Model for Highway Accident Prediction: A Case Study of Erzurum, Turkey. *Promet – Traffic&Transportation*. 2015;27(3): 217-25.

[4] Linchao L, Fratrović T, Jian Z, Bin R. Traffic Speed Prediction for Highway Operations Based on a Symbolic Regression Algorithm. *Promet – Traffic&Transportation*. 2017;29(4): 433-441.

[5] Tseng F, Hsueh J, Tseng C, Yang Y, Chao H, Chou L. Congestion Prediction with Big Data for Real-Time Highway Traffic. *IEEE Access*. 2018;6: 57311-57323. DOI: 10.1109/ACCESS.2018.2873569

[6] Zhao H, Xia J, Li F, Li Z, Li Q. A Peak Traffic Congestion Prediction Method Based on Bus Driving Time. *Entropy.* 2019;21(709). DOI: 10.3390/e21070709

[7] Transportation Research Board. *The Highway Capacity Manual 2010 (HCM2010).* The National Academies of Science, United States; 2010.

[8] Xing Y, Ban XJ, Liu X, Shen Q. Large-Scale Traffic Congestion Prediction Based on the Symmetric Extreme Learning Machine Cluster Fast Learning Method. *Symmetry*. 2019;11(6): 730. DOI: 10.3390/sym11060730

[9] Huang D, Deng Z, Wan S, Mi B, Liu Y. Identification and Prediction of Urban Traffic Congestion via Cyber-Physical Link Optimization. *IEEE Access*. 2018;6: 63268-63278.

[10] Feifei H, Yan XD, Liu Y, Ma LA. Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index. *Procedia Engineering*.

2016;137: 425-433. DOI: 10.1016/j.proeng.2016.01.277

[11] Nguyen DB, Dow CR, Hwang SF. An Efficient Traffic Congestion Monitoring System on Internet of Vehicles. *Wireless Communications and Mobile Computing*. 2018. DOI: 10.1155/2018/9136813

[12] Zhang H, Shi B, Zhuge C, Wang W. Detecting Taxi Travel Patterns using GPS Trajectory Data: A Case Study of Beijing. *KSCE Journal of Civil Engineering*. 2019;23: 1797-1805. DOI: 10.1007/s12205-019-0580-6

[13] Dawei N, Zhichao M, Hai W, Weibo Y, Wendong Z, Xiao G. A cross traffic estimate method for high speed networks. *Proceedings of IEEE 14th International Conference on Communication Technology, 2012, Chengdu;* 2012.

[14] Liu Y, Feng X, Wang Q, Zhang H, Wang X. Prediction of Urban Road Congestion Using a Bayesian Network Approach. *Procedia - Social and Behavioral Sciences.* 2014;138: 671-678. DOI: 10.1016/j.sbspro.2014.07.259

[15] Lee J, Hong B, Lee K, Jang YJ. A Prediction Model of Traffic Congestion Using Weather Data. *Proceedings of IEEE International Conference on Data Science and Data Intensive Systems, 2015, Sydney*; 2015.

[16] Liu Y, Wu H. Prediction of Road Traffic Congestion Based on Random Forest. *Proceedings of 10th International Symposium on Computational Intelligence and Design (ISCID), 2017, Hangzhou*; 2017.

[17] Mondal MA, Rehena A. Intelligent Traffic Congestion Classification System using Artificial Neural Network. *Proceedings of Second International Conference on Advanced Computational and Communication Paradigms (ICACCP-2019);* 2019.

[18] Elleuch W, Wali A, Alimi AM. 2017. Intelligent Traffic Congestion Prediction System Based on ANN and Decision Tree Using Big GPS Traces. *Advances in Intelligent Systems and Computing*. 2017; 557. DOI: 10.1007/978-3-319-53480-0_47

[19] Liu Z, Li Z, Wu K, Li M. Urban Traffic Prediction from Mobility Data Using Deep Learning. *IEEE Network*. 2018;32(4): 40-46. DOI: 10.1109/MNET.2018.1700411

[20] Zhang S, Yao Y, Hu J, Zhao Y, Li S, Hu J. Deep Autoencoder Neural Networks for Short-Term Traffic Congestion Prediction of Transportation Networks. *Sensors (Basel).* 2019;19(10): 2229. DOI: 10.3390/s19102229

[21] Zhang T, Liu Z, Cui Z, Leng J, Xie WH, Zhang L. Short-Term Traffic Congestion Forecasting Using Attention-Based Long Short-Term Memory Recurrent Neural Network. *Proceedings of International Conference on Computational Science*; 2019.

[22] Zhou X, Dong P, Xing P, Sun P. Learning Dynamic Factors to Improve the Accuracy of Bus Arrival Time Prediction via a Recurrent Neural Network. *Future Internet. 2019*;11: 247. DOI: 10.3390/fi11120247

[23] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 1994;5(2): 157-166. DOI: 10.1109/72.279181

[24] Cho KH, Merrienboer BV, Bahdanau D, Bengio Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, October, 2014, Doha, Qatar*; 2014.

[25] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. *Proceedings of ICASSP 2013, May, 2013, Vancouver, Canada*; 2013.

[26] Sutskever I, Vinyals O, Quoc VL. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*. 2014;4: 3104-3112.

[27] Sun S, Chen J, Sun J. Traffic congestion prediction based on GPS trajectory data. *International Journal of Distributed Sensor Networks*. 2019;15(5). DOI: 10.1177/1550147719847440

[28] Chen M, Yu G, Chen P, Wang YP. Traffic Congestion Prediction Based on Long-Short Term Memory Neural Network Models. *Proceedings of 17th COTA International Conference of Transportation Professionals*; 2017.

[29] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans.* 2017;39(12): 2481-2495. DOI: 10.1109/TPAMI.2016.2644615

[30] Liu X, Deng Z, Yang Y. Recent progress in semantic image segmentation. *Artif. Intell. Rev.* 2019;52(2): 1089-1106. DOI: 10.1007/s10462-018-9641-3

[31] Kurniawan J, Syahra SG, Dewa CK. Traffic Congestion Detection: Learning from CCTV Monitoring Images using Convolutional Neural Network. *Procedia Computer Science*. 2018;144: 291-297. DOI: 10.1016/j.procs.2018.10.530

[32] Ma X, Dai Z, He Z, Ma J, Wang Y, Wang Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*. 2017;17(4): 818. DOI: 10.3390/s17040818

[33] Zahid M, Chen Y, Jamal A, Memon MQ. Short Term Traffic State Prediction via Hyperparameter Optimization Based Classifiers. *Sensors*. 2020;20(3): 685. DOI: 10.3390/s20030685

[34] Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 1962;60(1): 106-154.

[35] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*. 2012;25(2). DOI: 10.1145/3065386

[36] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. 1958;65(6): 386-408.

[37] Ruder S. An overview of gradient descent optimization algorithms. *arXiv*. 2016; arXiv:1609.04747.

[38] Matthew DZ. AdaDelta: An adaptive learning rate method. *arXiv*. 2012; arXiv:1212.5701v1.

[39] Kingma DP, Ba JL. Adam: A method for stochastic optimization. *arXiv*. 2014; arXiv:1412.6980v9.

[40] Rumelhart D, Hinton G, Williams R. Learning representations by back-propagating errors. *Nature.* 1986;323: 533-536. DOI: 10.1038/323533a0

[41] James LM, Rumelhart DE. Schemata and Sequential Thought Processes in PDP Models. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. 1987; 7-57.