Modeling Urban Traffic Data through Graph-Based Neural Networks

Viviana Pinto[1], Alan Perotti[1,2], and Tania Cerquitelli[3]

[1] aizoOn Technology Consulting, Turin, Italy
[2] Institute for Scientific Interchange, Turin, Italy
[3] Department of Control and Computer Engineering, Politecnico di Torino, Italy

Abstract. The use of big data in transportation research is increasing and this leads to new approaches in modeling the traffic flow, especially for what concerns metropolitan areas. An open and interesting research issue is city-wide traffic representation, correlating both spatial and time patterns and using them to predict the traffic flow through the whole urban network. In this paper we present a machine learning based methodology to model traffic flow in metropolitan areas with the final aim to address short-term traffic forecasting at various time horizons. Specifically, we introduce an ad-hoc neural network model (GBNN, Graph Based Neural Network) that mirrors the topology of the urban graph: neurons corresponds to intersections, connections to roads, and signals to traffic flow. Furthermore, we enrich each neuron with a memory buffer and a recurrent self loop, to model congestion and allow each neuron to base its prediction on previous local data. We created a GBNN model for a major Italian city and fed it one year worth of fine-grained real data. Experimental results demonstrate the effectiveness of the proposed methodology in performing accurate lookahead predictions, obtaining 3% and 16% MAAPE error for 5 minutes and 1 hour forecasting respectively.

Keywords: Traffic · Neural Networks · Transportation.

1 Introduction

Around 50% of the global population lives in metropolitan areas, and this is likely to grow to 75% by 2050 [13]. Mobility within a wide metropolitan area is a complex system that needs to be modeled properly in order to predict the traffic flow and the consequent impact of congestions. Nowadays lots of data are available from detectors such as city cameras or induction loops: these devices can generally register the number of vehicles transitioned in an interval of time and their speeds. Furthermore these data can be integrated with information sourced from crowds, e.g. using mobile phone applications or sensors embedded into vehicles such as GPS. Data-driven approaches can be exploited to extract

The work presented in this paper is part of the SETA project, funded by the European Commission as part of the Horizon 2020 programme under contract 688082.

useful knowledge from those data and to effectively support short-term traffic forecasting. Traffic management systems can be trained on topological data and traffic measurements in order to estimate and predict the traffic flow. In this paper we use Neural Networks (NN) to model the traffic flow and to make short-term traffic forecasting. We present an original NN model based on the topology of the city (Graph Based Neural Network - GBNN), and we integrate topological and traffic data for a major Italian city (Turin) and we feed them to our GBNN model to perform traffic flow prediction.
The paper is outlined as follows. In section 2 we present the state-of-the-art in traffic modeling. In section 3 we discuss the GBNN model, including the general vision and the mathematical model, then we present experimental evaluation and results in section 4. In section 5, we end the paper with final remarks and directions for future work.

2 Related Work

In the last few years urban traffic data have received a great attention in different research areas such as transportation and machine learning. Transportation Research interest has been mainly focused on developing methodologies that can be used to model traffic characteristics such as volume, density and speed of traffic flows [7]. According to [15], most short-term traffic forecasting algorithms were built to function at a motorway, freeway, arterial or corridor level, not considering roads intersections at urban level. Predictions at a network level using data driven approaches remains a challenging task; the difficulty in covering a sufficient part of the road network by sensors, as well as the complex interactions in densely populated urban road networks, are among the most important obstacles faced in short-term traffic forecasting. In particular, NN have been widely applied to various transportation problems, due to their ability to work with massive amounts of multi-dimensional data, their modeling flexibility, and their learning and generalization ability: in [2] there is an early idea of application of NN to Transportation Engineering problems. A more structured idea of a possible application of NN in order to simulate and predict traffic flow can be found in [12], where a NN-based system identification approach is used to establish an auto-adaptive model for the dispersion of traffic flow on road segments or in [16], where a space-time delay NN is constructed to model autocorrelation of road traffic networks. Considering a section of freeway, a similar NN is built by [14] to model traffic flow and to obtain short-term prediction using real data; in particular, their work is focused on dealing with missing or corrupted data. Ma et al. [6] use traffic sensor data as input for a NN that predicts travel speeds, they consider a long period of time with high frequency data on an expressway in Beijing. Zhu et al. [17] study traffic volumes of three adjacent intersections, observing that traffic flows of single intersections are significantly affected by the traffic flows at the adjacent intersections. Polson et al. [11] focus on the use of road sensor data on a corridor in Chicago during a special event in order to study data representing a congestion; they develop a deep learning architecture that combines a linear model with a NN for the purpose of identify spatio-temporal relations among predictors and model nonlinear relations as well.

2.1 Neural Networks (NN)

Artificial neural networks [9] are computational paradigms composed by a series of interconnected processing elements that operate in parallel. The most used learning algorithm is the backpropagation algorithm, proposed and popularized by Hinton et. al in 1986 [3]. A recurrent neural network (RNN) allows for connections from the output layer to the input layer (compared with traditional feed-forward networks, where connects feed only to subsequent layers). Because RNNs include loops, they can store information while processing new inputs. This memory makes them ideal for processing tasks where prior inputs must be considered (such as time-series data). In particular, in Nonlinear Auto Regressive with eXogenous inputs (NARX) networks [8] each recurrent link includes a delay unit, so that the activation value of an output neuron O at time t is applied to an input neuron I at time t + 1.

3 Graph-based Neural Network Model

In order to study the urban network, we model the city as a graph in which each junction corresponds to a node and each road to an edge. Turin's topological structure allows for this kind of model with small information loss due to the general straightness of roads. For the few curved roads, we choose to simplify the graph by representing them as straight edges. The resulting urban graph is strongly connected and, as we do not represent overpasses and underpasses, also planar. Ideally the graph should be direct with some multiple edges and loops; however, we decided to consider only the main roads and use an undirected graph.
We first create a neural network that matches the city graph, so that each graph node, representing a road intersection, corresponds to a neuron and each edge, representing a road, corresponds to a connection between neurons. The traffic flow is then modeled as signal propagation within the network - on ingoing/outgoing connections between neurons. We consider the possibility that not all the cars approaching a junction are able to leave it in a single time frame - we enable local congestion by means of a recurrent connection on each neuron. Note that the system we are considering is not a closed system: flow mass can be lost or generated, according to detected data. This can happen because of vehicles entering/leaving the city, or simply being parked or put back into circulation during the day. Starting from the gathering of real data from traffic detectors, we build the city graph with associated times series of traffic flow and we design a NN based on these information. Then we feed the traffic data to the NN and we analyze the results. In particular, we can divide the data processing in four steps, as follows:

1. *Data gathering and preprocessing* $\rightarrow$ collection of data referred to the topology of the city, collection of data referred to city detectors position and registered traffic flow, handling of missing data, propagation of data from city detectors to city nodes.
2. *Data exploration and visualization* $\rightarrow$ analysis of frequent patterns in the behavior of traffic flow in a day-interval and a week-interval.
3. *Neural Network model design* $\rightarrow$ creation of the GBNN model, definition of neuron-level and network-level parameters, learning procedure design.
4. *Model evaluation assessment* $\rightarrow$ performance analysis of the GBNN model.

3.1 Data gathering and preprocessing

Two different kinds of data are used in order to create the model. In the first phase we use topological data to shape the city graph; in the second phase we use data collected by road detectors to model traffic flow. We focus on the city

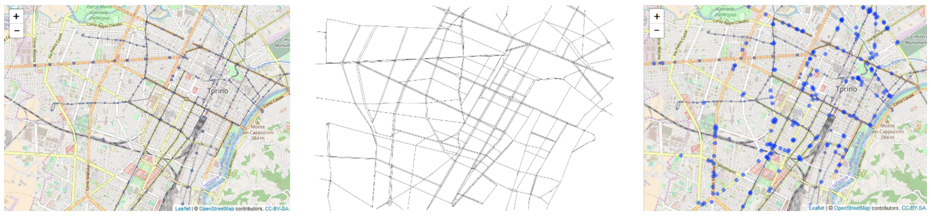

Fig. 1: Turin city map, road network, and detectors distribution.

of Turin, Italy, (Fig. 1 - left). We create the graph associated with the city using latitude-longitude data; we consider the main intersections and roads of the city center, thus obtaining a graph with 1074 nodes and 1874 edges (Fig. 1 - middle). The 201 considered detectors are positioned in the city as shown in Fig. 1 - right. Detectors data are given in time series with a granularity of five minutes in a period of one year (from June, 27th 2016 to June, 27th 2017) and the available data are the number of vehicles in a time interval and their average speed. We aggregate these data in a unique indicator, the flow. In order to use these data in the NN, the flow must be propagated from detectors to graph nodes (i.e. neurons of the NN), and we do it as follows:

$$f_i = \sum_{j=1}^{N} \frac{\hat{f}_j}{dist^2(i,j)}, \tag{1}$$

where f_i is the flow on vertex/neuron i, $\hat{f}_j$ is the flow on detector j, N is the number of detectors, and *dist* is the euclidean distance between the node i and detector j. Turin city center area has 463 detectors, but we consider only detectors with almost complete data ($> 75\%$) with respect to time that are 201, in order to avoid as much as possible the artificial generation of data to fill the dataset. We filled each data gap with the average value of the last preceding and first following data-points.

3.2 Data exploration and visualization

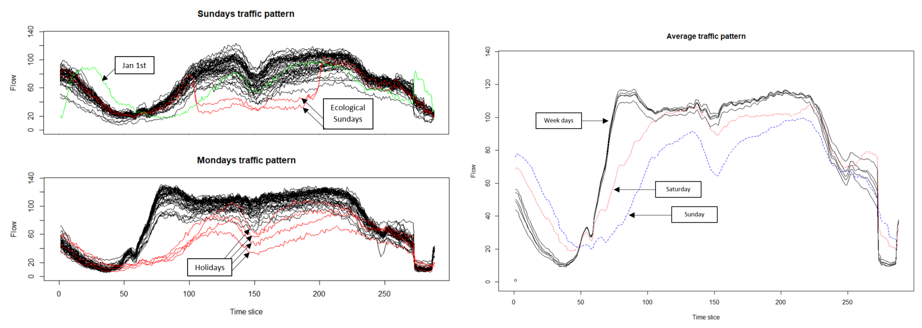

Fig. 2: Daily and weekly traffic patterns.

According to the previously presented set-up, we studied the distribution of the flow during the day. We found three different emerging patterns: one for the week days, one for Saturdays and one for Sundays, as it is shown in Fig. 2, where the average of all data for each week day and for each time slice is represented. Week days present a peak in morning hours that completely disappear during the weekend. Sundays present a big flow decrease during lunch time. Note that data present a behavior out of the path in late-evening hours. This is probably due to a technical constraint of the considered detectors.
Some interesting patterns emerge from the distributions of Sundays and Mondays. In particular, the flow distribution of the Sundays presents some exceptions, that are days out of the general pattern as shown in Fig. 2 - top left: the green line represents January 1st and we can see an inverted behavior in early morning hours, while during the day the total amount of flow is smaller than other Sundays in the year; the red lines represent March 5th and April 2nd, ecological Sundays, in which the cars were not allowed to circulate from 10am to 6pm, thus the line perfectly follows the pattern until 10am, then drops until 6pm and then resumes matching the general pattern. For what concerns Mondays, Fig. 2 - bottom left shows some days with patterns similar to Sundays, with a smaller amount of traffic and with no morning peak, this lines represent holidays.

3.3 Neural network model design

We create a recurring neural network able to perform short-term traffic prediction from available data. We associate each node of the urban graph, corresponding to a junction, to a neuron and each edge, corresponding to a road, to a neurons connection. Traffic flow is then modeled as propagating signal through the neural network. Since each neuron corresponds to a node of the urban graph, it is natural that neurons have multiple outgoing connections representing the outgoing roads of the junction. Each neuron has numerous ingoing connections, which are controlled by a delay layer representing the time window management; inputs are combined and an output is produced and then partitioned on the outgoing connections until the connections are saturated; the exceeding flow is inserted again in the neuron through the self-loop. The flow on this loop represents the traffic congestion on a junction. The starting urban graph is composed by nodes and edges, such that for each graph node i, $i = 1, 2, \ldots, N$, there exists a stack of previsions $f_i(t)$, $t \in [0, T]$, with flow data for each time interval. We build a NN with a neuron for each graph node and a connection for each graph edge. Furthermore, for each neuron i we create an artificial self loop (i, i), and for each connection (i, j) we set a maximum capacity $C_{i,j}$. In this work, we set this capacity to a flow value of 100, which is the third quartile of the flow distribution. Consider a time window of length $\tau + 1$ and let $F_i(t) = [f_i(t), f_i(t-1), \ldots, f_i(t-\tau)]^T$ be the vector of the time series associated with node i. Let M be the number of in-going edges of node i and K be the number of out-going edges from node i. For each node i a weight matrix $W^i(t)$ is defined, with K rows and τ columns. We initialized each weight matrix with exponential decay and Gaussian noise ($\sim \mathbf{N}(0, 1)$) as follows:

$$W^k(0) = \{w^k_{ij}(0)\}_{ij} = \left\{ \frac{1}{2^j} + noise \right\}_{ij}, \quad k = 1, 2, \ldots, N \tag{2}$$

so that each node initially routes the traffic equally to adjacent nodes (since $w_{i,j}$ does not depend on i), and weighing recent history more than previous one (as $w_{i,j}$ halves for every previous timestamp). We use the matrix $W^i(t)$ to estimate the K out-going flows from node i at time $t + 1$.

$$W^i(t) \cdot F_i(t) = \begin{bmatrix} w^i_{11} & w^i_{12} & \ldots & w^i_{1\tau} \\ w^i_{21} & w^i_{22} & \ldots & w^i_{2\tau} \\ \ldots & \ldots & \ldots & \ldots \\ w^i_{K1} & w^i_{K2} & \ldots & w^i_{K\tau} \end{bmatrix} \cdot \begin{bmatrix} f_i(t) \\ f_i(t-1) \\ \ldots \\ f_i(t-\tau-1) \end{bmatrix} = \begin{bmatrix} \tilde{f}_{i1}(t+1) \\ \tilde{f}_{i2}(t+1) \\ \ldots \\ \tilde{f}_{iK}(t+1) \end{bmatrix} = \tilde{\mathcal{F}}_i(t+1), \tag{3}$$

where $\tilde{f}_{ij}(t+1)$ is the estimated flow at time $t+1$ on edge (i, j) and $\tilde{\mathcal{F}}_i(t+1)$ is the vector of the estimated out-going flows from node i.

$$\begin{cases} \tilde{f}_i(t) = \sum\limits_{j=1}^{M} \tilde{f}_{ji}(t) + \tilde{f}_{ii}(t), \\ \tilde{f}_{ii}(t) = \max\left\{ \sum\limits_{j=1}^{K} \left(\tilde{f}_{ij}(t) - C_{ij} \right), 0 \right\}. \end{cases} \tag{4}$$

Formally, the estimated flow on node i at time $(t+1)$ is the sum of the estimated flows coming from its M in-going edges plus the already existing flow on its self-loop. The flow on the self-loop is the sum of the flows that cannot exit on edge

(i,j) because the maximum capacity C_{ij} has already been reached.
For each node i we define an error $E_i(t)$ as follows

$$E_i(t) = f_i(t) - \tilde{f}_i(t). \tag{5}$$

Then we partition this error on the in-going edges on node i proportionally to the in-going flows as follows

$$E_{ji}(t) = \frac{\tilde{f}_{ji}(t-1) \cdot E_i(t)}{\sum_{j=1}^{M} \tilde{f}_{ji}(t-1)}. \tag{6}$$

We define the error $E^i(t)$ as the vector of the outgoing error from node i at time t.

$$E^i(t) = [E_{i1}(t), \ldots, E_{iK}(t)]. \tag{7}$$

Using the error vector $E^i(t)$, we can update backwards the matrix $W^i(t+1)$ for each node i as follows. Moreover, we add a parameter η, the learning rate.

$$W^k(t+1) = \{w_{ij}^K(t+1)\}_{ij} \quad = \left\{ \frac{w_{ij}^K(t) + \eta \cdot E^K(t) \cdot f_k(t)}{\sum_{i,j} w_{ij}^K(t+1)} \right\}_{ij}. \tag{8}$$

3.4 Model evaluation assessment

In order to evaluate the predictive ability of the GBNN model, we adopted MAAPE, Mean Arctangent Absolute Percentage Error as proposed by [5]. MAPE (Mean Average Percentage Error) is the most widely used measure of forecast accuracy, but it is asymmetric and afflicted by the problem of the division by zero. To avoid this problem but to preserve the philosophy of MAPE, the MAAPE is introduced. It is defined as follows:

$$\text{MAAPE} = \frac{1}{N} \sum_{t=1}^{N} \arctan\left(\frac{\mid \text{predicted} - \text{expected} \mid}{\text{expected}} \right). \tag{9}$$

The error belongs to $[0, \frac{\pi}{2}]$. In order to get a percentage value, we rescale the MAAPE by multiplying by $\frac{2}{\pi}$, such that now the error belongs to $[0,1]$.

4 Experimental Results

We implemented our GBNN model in Java. We set the hyperparameters as follows: the dimension of the considered time windows (memory size) is 4, the maximum capacity of the edges is 100.00 for all links in the graph, the learning rate is 10^{-4}, the maximum time horizon for predictions is 12 (that is prediction 1 hour ahead), the number of train iterations is 40. We proceeded to test the GBNN on each week day separately, using all days except one as training set and considering the remaining day as validation. We use the trained network to predict the traffic flow at different time horizons: that is, using the validation data

Day	Time horizons			
	5m (%)	10m (%)	30m (%)	1h (%)
Monday	2.89	4.05	9.43	15.85
Tuesday	2.95	3.99	9.24	15.62
Wednesday	2.82	3.90	9.09	15.33
Thursday	2.77	3.91	9.48	16.10
Friday	2.60	3.63	8.91	15.56
Saturday	2.39	2.96	6.78	12.13
Sunday	2.71	3.17	7.18	13.23
Average	**2.73**	**3.66**	**8.59**	**14.83**

Table 1: Table of MAAPE errors at different horizons

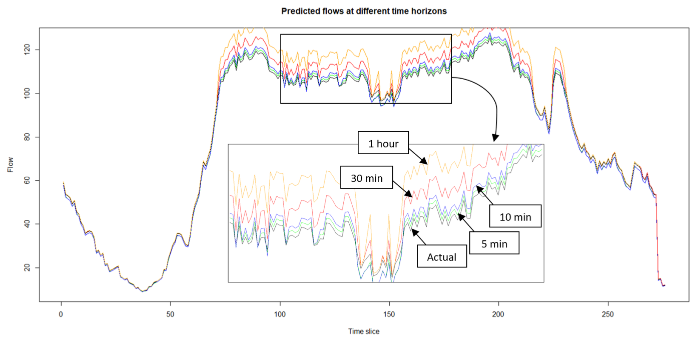

Fig. 3: Flow predictions analysis at different time horizons.

at time t to predict the traffic flow at time $t+k$ minutes, with $k \in \{5, 10, 30, 60\}$.

We obtain very small errors for predictions 5 minutes ahead and the error is still under 10% for predictions 30 minutes ahead. GBNN is able to predict traffic flow 1 hour ahead with an error around 15%. The results are almost constant for each day-based GBNN and for each time horizon, as shown in Table 1. We select a random neuron to analyze predicted flow values at different time horizons qualitatively. Fig. 3 shows the actual flow and the predicted flows for the validation day on Mondays set. GBNN performs excellent prediction for the first and the last part of the day, while it is subjected to a bigger error during the middle hours. For our experiments, we used a Dell Precision M3800 (Intel Core i7-4712HQ, 16GB RAM). Roughly, each training iteration requires 0.05 seconds for each day in the training set (288 time slices per day). We performed random search and grid search in order to find the optimal values for learning rate and memory size, that are the most incisive hyperparameters (see Fig. 4). In the first case, the lowest MAAPE was associated with a learning rate of 10^{-4}: concerning memory size, the value 4 (that is, to base the prediction upon the previous 20 minutes of data) was correlated with the lowest cross-horizon MAAPE. The shaded area in Fig. 4 represents the interval of values which minimise the average error. We decide to consider the trivial model, in which predictions are made by repeating the value of the known flow, to compare the results because classical statistical methods do not fit our urban traffic representation: incomplete data and instable distribution undermine the stability of statistical methods. We decide not to compare our model to an agent-based one because the approach completely differs: we consider the traffic mass as a whole, not as a set of individual agents. Moreover, there is no existing neural network model in literature which consider the entire urban network or the bidirectional traffic flow. The comparison of the GBNN model and the trivial one is shown in Fig. 5 for 5 minutes ahead predictions (left) and for 1 hour ahead predictions (right), on the validation day for a randomly selected node. The top panels show the comparison of real flow, flow predicted by the trivial model (red line) and flow predicted by GBNN (blue line) for a time horizon of 5 minutes (left) and 1 hour (right). The bottom panels display the error comparison for the models: red line for trivial model, black line for GBNN.

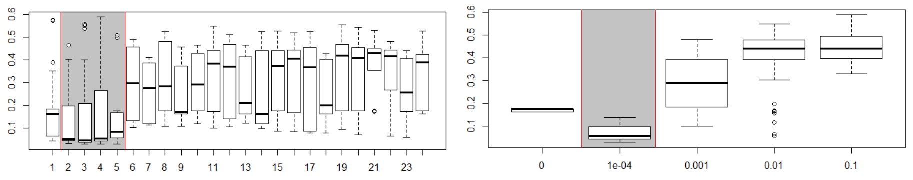

Fig. 4: MAAPE error analysis w.r.t. memory size (left) and learning rate (right).

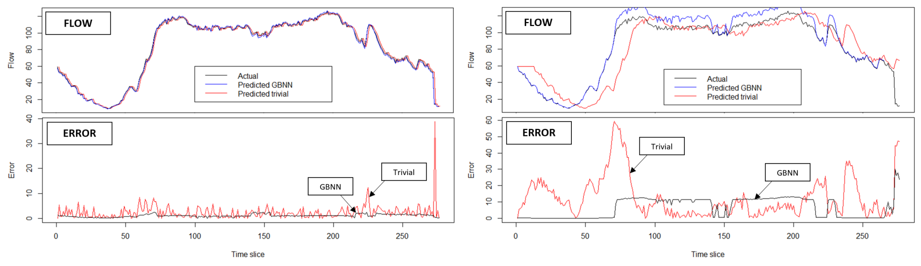

Fig. 5: Flows and errors comparison

5 Conclusions and Future Work

In this paper we have proposed an innovative neural network model, whose topology mirrors an urban graph and where each node is enriched with a recurrent connection in order to capture the temporal behavior. The resulting system is able to model the whole urban traffic flow, to take into account both space and

time patterns in the data, and consequently to make network-wide traffic flow predictions. The prediction error is as low as 3% and 16% for 5 minutes and 1 hour traffic forecasting respectively.
Future directions of work will focus on implementing peculiarities of urban road network that we omitted for the initial model, such as one-way street and roundabouts.

References

1. M. Chowdhuri and A.W. Sadek, "Advantages and limitations of artificial intelligence", Trans. Res. B., vol. E-C168, 6–8, 2012.
2. A. Faghri and J. Hua, "Evaluation of artificial neural network applications in transportation engineering", Transport. Res. Rec., vol. 1358, 71–80, 1992.
3. D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation", MIT Press, 1986.
4. M.G. Karlaftis and E.I. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights", Transport. Res. C-Emer., vol. 19, 387–399, 2011.
5. S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts", J. Forecasting, vol. 32, 669–679, 2016.
6. X. Ma, Z. Tao, Y. Wang, H. Yu and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data", Transport. Res. C-Emer., vol. 54, 187–197, 2015.
7. F. Moretti, S. Pizzuti, S. Panzieri and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensamble hybrid modeling", Neurocomputing, vol. 167, pp. 3–7, 2015.
8. Hava T. Siegelmann, Bill G. Horne, and C. Lee Giles. "Computational capabilities of recurrent NARX neural networks", IEEE Transactions on Systems, Man, and Cybernetics, Part B, 27(2):208215, 1997.
9. S. Haykin, "Neural networks: a comprehensive foundation", Prentice Hall, 1999.
10. F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain", Psychol. Rev., vol. 65, 386-408, 1958.
11. N.G. Polson and V.O. Sokolov, "Deep learning for short-term traffic flow prediction", Transport. Res. C-Emer., vol. 79, 1–17, 2017.
12. F. Qiao, H. Yang and W.H.K. Lam, "Intelligent simulation and prediction of traffic flow dispersion", Transport. Res. B-Meth., vol. 35, 843–863, 1999.
13. United Nations, Department of Economic and Social Affairs, Population Division, "World urbanization prospects: the 2014 revision", 2014.
14. J.W.C. van Lint, S.P. Hoogendoorn and H.J. van Zuylen, "Accurate freeway travel time prediction with state-space neural network under missing data", Transport. Res. C-Emer., vol. 13, 347–369, 2005.
15. E.I. Vlahogianni, M.G. Karlaftis and J.C. Golias, "Short-term traffic forecasting: where we are and where we're going", Transport. Res. C-Emer., vol. 43, 3–19, 2014.
16. J. Wang, I. Tsapakis and C. Zhong, "A space-time delay neural network model for travel time prediction", Eng. Appl. Artif. Intel., vol. 52, 145–160, 2016.
17. J.Z. Zhu, J.X. Cao and Y. Zhu, "Traffic volume forecasting based on radial basis fuction neural network with the consideration of traffic flows at the adjacent intersections", Transport. Res. C-Emer., vol. 47, 139–154, 2014.