ISSN 1060-992X, Optical Memory and Neural Networks, 2021, Vol. 30, No. 1, pp. 1–10. © Allerton Press, Inc., 2021.

Short-Term Traffic Data Forecasting: A Deep Learning Approach

A. A. Agafonov*

Samara National Research University, Samara, 443086 Russia

**e-mail: ant.agafonov@gmail.com*

Received August 7, 2020; revised December 1, 2020; accepted December 7, 2020

Abstract—Accurate and timely traffic forecasting plays an important role in the development of intelligent transportation systems (ITS). Traffic data are the main information source for various tasks solved as part of the ITS, including traffic management, urban planning, route guidance, and others. Due to the spatial and temporal nonlinearity and complexity of traffic flow, traffic forecasting problem remains a subject of research. In this paper, we design a deep-learning framework that combines convolution operations on graph data with recurrent neural networks to solve the short-term traffic data forecasting problem. The proposed model takes into account recent, daily, and weekly periodic time series to capture different patterns in traffic flow. The experimental study of the model conducted on publicly available real-world datasets shows that the proposed model outperforms other baseline methods.

Keywords: traffic forecasting, graph neural network, spatial convolution, temporal convolution, deep learning, LSTM

DOI: 10.3103/S1060992X21010021

1. INTRODUCTION

Intelligent transportation systems (ITS) play an increasingly important role in urban life. ITS can solve tasks in different fields of transportations systems, including transportation management and urban planning, public transport management, development of routing services provided for road users, and so on. Widespread use of ITS allows reducing the traffic congestion level in an urban environment, emission level, accident risk, and increase safety and reliability of transportation infrastructure [1].

By application areas, various fields in ITS can be classified as follows [2]:

(1) Advanced Traveler Information System provides information about traffic congestions, arrival times of public transport, navigation services, and others [3, 4].

(2) Advanced Public Transportation System considers public transportation management and scheduling problems taking into account the congestion level in the transportation network.

(3) Advanced Traffic Management System provides instruments to monitor and control traffic flow, including real-time management.

(4) Emergency Management System provides services in emergency conditions like accidents.

It is clear, that all these systems use traffic flow information as a base data source to solve their practical tasks and improve ITS service quality as a whole. As a result, traffic flow forecasting remains one of the most popular research tasks in the transportation field.

The short-term traffic forecasting problem considered in this paper can be described as follows: given current and historical traffic flow observations and topology of a road network, predict variables of traffic flow (namely speed, volume, or density) up to one hour ahead. The main challenges in solving this problem are determined by complex spatial-temporal nonlinear dependencies in traffic data. Recent advances in traffic forecasting have been achieved using various machine learning methods and algorithms, including recurrent learning networks and deep learning approaches. Despite the large number of papers devoted to this problem, the development of deep learning models for traffic forecasting remains a subject of research.

In this paper, we propose a novel deep learning model that combines convolution on graphs with recurrent neural networks. The proposed model uses a feature vector that takes into account recent, daily, and weekly periodic time series to capture different patterns in traffic flow. The model is evaluated with the real-world publicly available traffic data, showing that the proposed model outperforms other baseline methods.

The remainder of the paper is organized as follows. In Section 2, we present related works in the traffic forecasting research field. In Section 3, we give several definitions and formulate the problem. Details of the proposed method and architecture of the proposed model are presented in Section 4. Experimental results are provided in Section 5. Finally, we present conclusions and discuss future research.

2. RELATED WORK

A review of the short-term traffic forecasting problem can be found in [5, 6]. The authors overviewed recent advances and main challenges, classified existing approaches according to various criteria, including the data collection technologies, the predicted characteristics of traffic flows, the used models and algorithms, etc., and described unsolved problems.

Earlier research papers are mostly based on classical statistical methods to predict traffic flow. Widely-used time-series models include ARIMA models [7], seasonal ARIMA models [8], vector autoregression models [9], etc.

Later approaches used machine learning methods and intelligence-based approaches to develop traffic forecasting algorithms. The widespread use of such approaches became possible due to the active development of high-performance computing systems, including platforms with graphs processing units (GPU). These methods including neural networks [10, 11], support vector regression [12], k-nearest neighbors [13]. These methods can efficiently process time series data, but cannot handle the spatial-temporal dependencies in traffic data.

Recent studies use deep learning tools to efficiently process complex nonlinear dependencies in traffic data and analyze traffic patterns. In [14], the authors presented Long Short-Term Memory (LSTM) neural networks for traffic forecasting. Gated Linear Unit (GRU) neural networks were investigated in [15]. Other works [16, 17] also demonstrated the advantages of LSTM models for traffic prediction. However, such approaches do not consider spatial correlations in traffic data.

Inspired by the successful application of convolution operators in pattern recognition and image analysis, several studies investigated graph convolution neural networks [18, 19]. In [20], the authors transformed graph data into grid-like structures to use regular convolutional operations. In [21], the authors presented semi-supervised learning on graph-structured data for classification tasks. Spatio-temporal graph convolutional networks were proposed in [22] to tackle the time-series prediction problem in the traffic domain. In [23], the authors proposed combine graph convolution operation with LSTM neural networks for traffic learning and forecasting. However, the authors considered only the recent traffic flow data and forecasted the graph signals in the subsequent one step. The attention-based graph convolution neural network consisted of three components to model recent, daily and weekly periodic traffic patterns was proposed in [24]. In [25], the authors also used an attention mechanism for traffic speed forecasting. Several variants of LSTMs with a graph convolution block under the encoder-decoder framework to model spatial-temporal dependencies were implemented in [26].

In this paper, we propose a hybrid model that uses convolutional operators on graph-structured data in spatial and temporal domains to capture the spatial-temporal dependencies in traffic data with LSTM units for traffic forecasting in 60 min ahead. The proposed model takes into account recent, daily, and weekly periodic time series to capture different patterns in traffic flow.

3. METHODOLOGY

This section briefly describes the traffic forecasting problem, and key concepts used to perform spatial-temporal convolutions on graph-structured data.

3.1. Problem Statement

Traffic forecasting problem aimed at estimating the traffic characteristics in a certain time (or time interval) in the future using current and historical traffic flow observations and topology of the road network.

Introduce the following notations.

We consider a road network as a directed graph $G = (V, E, W)$ with nodes V and edges E, the number of nodes is $N = |V|$. Depending on data, each node can represent a road intersection, a road link, or a road sensor. $\boldsymbol{W} \in R^{N \times N}$ denotes the weighted adjacency matrix of the graph G.

Let $x_t^i \in R$ be the observed variable of traffic flow for a node $i \in V$ at a time moment t. The set of observed traffic flow variables for all nodes at time t denote as:

$$\mathbf{X}_t = \left(x_t^1, x_t^2, \ldots, x_t^N\right) \in \mathbb{R}^N. \tag{1}$$

The value of all traffic flow variables over τ time intervals denote as

$$\mathbf{X} = \left(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\tau\right)^T \in \mathbb{R}^{N \times \tau}. \tag{2}$$

The predicted value of traffic flow variables at future time t denote as

$$\widehat{\mathbf{X}}_t = \left(\hat{x}_t^1, \hat{x}_t^2, \ldots, \hat{x}_t^N\right) \in \mathbb{R}^N. \tag{3}$$

The traffic forecasting problem can be defined as follows:

Given a graph $G = (V, E, W)$ *and a sequence* $\mathbf{X}$ *of observed historical traffic flow data, predict the value of traffic flow variables* $\widehat{\mathbf{X}}_t$ *over the next* T_{ph} *time intervals*:

$$\widehat{\mathbf{X}}_{t+1}, \widehat{\mathbf{X}}_{t+2}, \ldots, \widehat{\mathbf{X}}_{t+T_{ph}} = f(\mathbf{X}; G), \tag{4}$$

where t is the current time.

In the next subsections, we describe the main concepts of the proposed hybrid spatial-temporal graph convolutional neural network.

3.2. Convolution in Spatial Domain

Convolution in the spatial domain allows processing localized spatial dependencies presented in road networks. Traditional convolutional operators that process 2D or 3D regular grid-based data are clearly not suitable to handle graph-based traffic data. To tackle this problem, the generalized convolutional operators based on the spectral graph theory was developed [19].

A common approach to define the convolutional operator on graph data is based on the adjacency matrix. Consider the Laplacian matrix of the graph:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

In normalized form, it can be defined as follows:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{\frac{1}{2}},$$

where $\mathbf{I}$ is a unit matrix, $\mathbf{D}$ is the diagonal matrix, consisting of node degrees:

$$\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}.$$

The eigenvalue decomposition of the Laplacian matrix can be defined as follows:

$$\mathbf{L} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T,$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix, $\mathbf{U}$ is the Fourier basis.

Using this notation, the convolutional operator on graph-based data is defined as the result of multiplying the signal $x \in \mathbb{R}^N$ on a graph with the kernel g_θ [27]:

$$g_\theta * x = g_\theta(\mathbf{L}) x = g_\theta\left(\mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T\right) x = \mathbf{U} g_\theta(\boldsymbol{\Lambda}) \mathbf{U}^T x.$$

Straight calculation of the convolution operation can be a computationally complex task, especially for large-scale graphs. To localize the filter g_θ, reduce the number of parameters, and decrease the computation time, it was proposed to generalize graph convolution using the Chebyshev polynomials [27]:

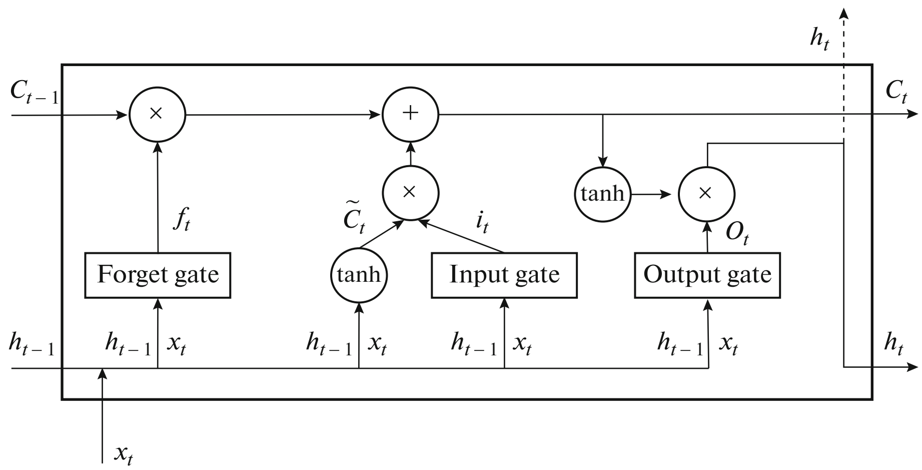

Fig. 1. Structure of the LSTM cell.

$$g_\theta * x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x,$$

where the θ_k parameter is a vector of polynomial coefficients, $\tilde{\mathbf{L}} = \frac{2}{\lambda_{max}}\mathbf{L} - I$ is the scaled Laplacian matrix, λ_{max} is the maximum eigenvalue of the Laplacian matrix. The Chebyshev polynomials are recursively defined as follows:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),$$
$$T_0(x) = 1, \; T_1(x) = x.$$

3.3. Convolution in Temporal Domain

To merge information from different time slices and capture temporal correlations in traffic data, after the graph convolution in the spatial domain we apply the convolution operation in the temporal dimension. Convolution in the temporal domain is performed on grid-based data using the standard convolutional operator. Calculations on the rth layer of the neural network can be described as follows:

$$x^r = ReLU\left(\Phi * \left(ReLU\left(g_\theta * x^{r-1}\right)\right)\right),$$

where * is the convolution operation, Φ is the parameter of the convolutional kernel in the temporal dimension, x is the original input time series, or the result of calculations on the previous layer of the neural network, $ReLU$ is the Rectified Linear Unit activation function.

3.4. Long Short-Term Memory (LSTM)

A long short-term memory (LSTM) neural network is a special type of the recurrent neural network (RNN) [28] with the ability to transfer a state of the cell from the previous time step to the next. LSTM neural network can be used to handle long-term dependency relations, especially when handling sequential data.

Figure 1 schematically shows the structure of the LSTM cell. In the schema, x is the input value, C is the state of the cell, h is the output value, t is the time step.

In the next section, we describe the proposed model.

4. PROPOSED MODEL

In this section, we present the hybrid deep learning model that combines spatial-temporal graph convolution layers with LSTM operators. Firstly, we describe the input data of the model (feature vector), that combines recent, daily, and weekly-periodic components of traffic data. Then, we present the architecture of the proposed model.

4.1. Input Feature Vector

To describe traffic flow in the road network, define a feature vector for all nodes of the graph G. According to (1), the set of observed traffic flow variables for all nodes at time t denoted as $\mathbf{X}_t \in \mathbb{R}^N$. In this paper, we propose to use the feature vector that takes into account daily and weekly-periodic components of traffic data.

Let the traffic data are received with the frequency q records per day. Denote the current time as t_0.The prediction horizon (the number of records to predict) denote as T_{ph}. To define the feature vector, consider the following time series:

(1) The recent time series combines the traffic data over the recent T_h time intervals:

$$\mathbf{X}^c = \left(\mathbf{X}_{t_0-T_h+1}, \mathbf{X}_{t_0-T_h+2}, \cdots, \mathbf{X}_{t_0}\right).$$

(2) The daily-periodic time series:

$$\mathbf{X}_d^{day\_past} = \left(\mathbf{X}_{t_0-d*q-T_h+1}, \mathbf{X}_{t_0-d*q-T_h+2}, \cdots, \mathbf{X}_{t_0-d*q}\right), \; d = \overline{1, T_d},$$
$$\mathbf{X}_d^{day\_shift} = \left(\mathbf{X}_{t_0-d*q-T_h+T_{ph}+1}, \mathbf{X}_{t_0-d*q-T_h+T_{ph}+2}, \cdots, \mathbf{X}_{t_0-d*q+T_{ph}}\right), \; d = \overline{1, T_d},$$

where T_d is the number of days to consider.

(3) The weekly-periodic time series:

$$\mathbf{X}_w^{week\_past} = \left(\mathbf{X}_{t_0-7*w*q-T_h+1}, \mathbf{X}_{t_0-7*w*q-T_h+2}, \cdots, \mathbf{X}_{t_0-7*w*q}\right), \; w = \overline{1, T_w},$$
$$\mathbf{X}_w^{week\_shift} = \left(\mathbf{X}_{t_0-7*w*q-T_h+T_{ph}+1}, \mathbf{X}_{t_0-7*w*q-T_h+T_{ph}+2}, \cdots, \mathbf{X}_{t_0-7*w*q+T_{ph}}\right), \; w = \overline{1, T_w},$$

where T_w is the number of weeks to consider.

Using the considering time series, the feature vector has the following form:

$$\mathbf{X}^{fv} = \left(\mathbf{X}^c, \left\{\mathbf{X}_d^{day\_past}\right\}_{d=\overline{1,T_d}}, \left\{\mathbf{X}_d^{day\_shift}\right\}_{d=\overline{1,T_d}}, \left\{\mathbf{X}_w^{week\_past}\right\}_{w=\overline{1,T_w}}, \left\{\mathbf{X}_d^{week\_shift}\right\}_{w=\overline{1,T_w}}\right) \in \mathbb{R}^{N\times F}, \quad (5)$$

where $F = 2T_d + 2T_w + 1$ is the number of time series.

4.2. Neural Network Architecture

In this paper, we propose a hybrid neural network model that combines spatial-temporal graph convolution operations with LSTM layers. Multiple blocks stacked together to extract a larger range of spatial-temporal correlations.

The architecture of the neural network is shown in Fig. 2.

The result of stacked spatial-temporal convolutions is concatenated with the input feature vector to perform LSTM processing. The output layer is a fully connected layer that is necessary to provide the required dimension of the output data.

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1. Datasets

We validate the proposed model on two publicly available highway traffic datasets PeMSD4 and PeMSD7. These datasets are collected by the Caltrans Performance Measurement System (PeMS) [29]. The datasets are aggregated into 5-min intervals from 30-s data samples, so each detector contains 288 points per day.

The PeMSD4 dataset was preprocessed in [24]. It contains data from 307 traffic sensors for 59 days. The PeMSD7 dataset was used in [22]. It contains data from 228 sensors for 44 days.

The main characteristics of the two datasets are presented in Table 1. Min-max normalization method was used to scale traffic data to [0, 1].

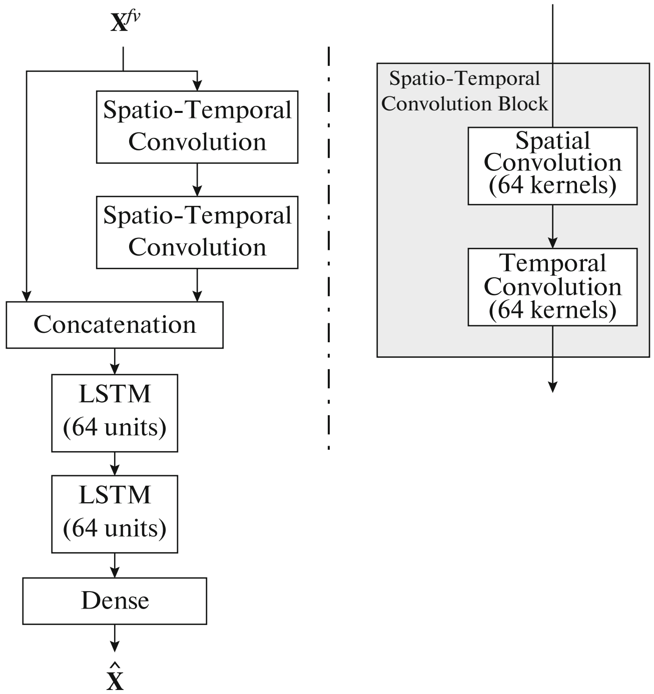

Fig. 2. Neural network architecture.

5.2. Settings

The proposed model was implemented using the Keras framework [30]. To create the feature vectors, we set the values of parameters as: $T_h = 12$ (recent data for the last hour), $T_d = 1$, $T_w = 1$. The prediction horizon value was chosen as $T_{ph} = 12$, so we aiming at predicting traffic flow data at one hour ahead.

The model parameters were learned using the training set, the best model was selected by the lowest error on the validation set. A comparison of models was carried out on the control set.

All experiments were conducted using the Linux cluster: Intel Core i7 9700K, 64GB RAM, GPU NVIDIA GeForce RTX 2080Ti.

The code has been released at https://github.com/ant-agafonov/stgcn-lstm.

5.3. Baseline Methods

We compare the proposed hybrid model with the following baseline methods:

• LSTM [28]: Long Short-Term Memory neural network;

• GRU: Gated Recurrent Unit network, a special RNN model;

• STGCN [22]: A spatial-temporal graph convolution network with gated linear unit activation mechanism;

• ASTGCN [24]: Attention-based spatial-temporal graph convolution network.

As the evaluation metrics, we use the following metrics: Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE).

Table 1. The characteristics of the PEMSD7 and PEMSD4 datasets

Dataset	Speed, km/h			Number of sensors	Number of records
	Min	Avg	Max		
PeMSD7	3.0	58.9	82.6	228	12672
PeMSD4	3.0	63.47	85.2	307	16992

Table 2. Performance comparison of different methods on the PEMSD7 and PEMSD4 dataset

Model	PEMSD7 (15/30/60 min)		PEMSD4 (15/30/60 min)	
	MAE	MAPE	MAE	MAPE
GRU	2.384/3.124/3.868	5.559/7.563/9.67	1.516/1.911/2.336	3.069/4.117/5.293
LSTM	2.329/3.035/3.803	5.446/7.385/9.604	1.536/1.905/2.336	3.084/4.096/5.292
STGCN	**2.258**/3.065/4.061	**5.293**/7.483/10.293	**1.452**/1.923/2.565	**2.899**/4.108/5.652
ASTGCN	2.469/3.186/4.017	5.793/7.764/10.136	1.744/2.122/2.664	3.634/4.698/6.052
Hybrid	2.261/**2.917/3.546**	5.305/**7.13/8.938**	1.468/**1.825/2.228**	2.95/**3.915/5.054**

5.4. Experiment Results

Tables 1 and 2 demonstrates the performance comparison results of the proposed Hybrid model and the baseline models on the datasets PeMSD7 and PeMSD4 respectively.

It can be seen, that the proposed model shows better results by selected criteria for almost all prediction horizon values. With the increase in the prediction horizon, the prediction error also increases because of error accumulations. In the prediction up to one hour ahead, the difference between the proposed model and the other baselines models are more significant.

Figures 3 and 4 shows the dependence of the criteria values for different methods on the prediction horizon (prediction intervals). Figure 3 shows the MAE values on the PeMSD7 dataset, Fig. 4 shows the MAPE values on the PeMSD4 dataset.

To estimate the model results, we also evaluate the prediction during the morning and evening rush hours for one sensor, for 30 and 60 min ahead, as shown in Figs. 5 and 6.

The proposed model is capable to respond to the dynamic changes in traffic data, but the performance quality decreases when the prediction horizon increases.

Average training time per epoch (in seconds) and testing time on one batch (traffic prediction up to one hour ahead for all sensors, in milliseconds) are shown in Table 3. It is should be noted, although all models were trained and tested using the same Linux cluster, they were implemented using different frameworks and libraries. Based on these experiments, we can conclude that all models can be used to predict traffic data in real-time.

CONCLUSIONS

In this paper, we consider the traffic data forecasting problem over one hour ahead. We propose a deep learning model that integrates spatial-temporal graph convolution with long short-term memory neural network. This approach allows us to capture complex nonlinear dependencies in traffic data. The proposed model takes into account recent, daily, and weekly periodic time series to capture different patterns

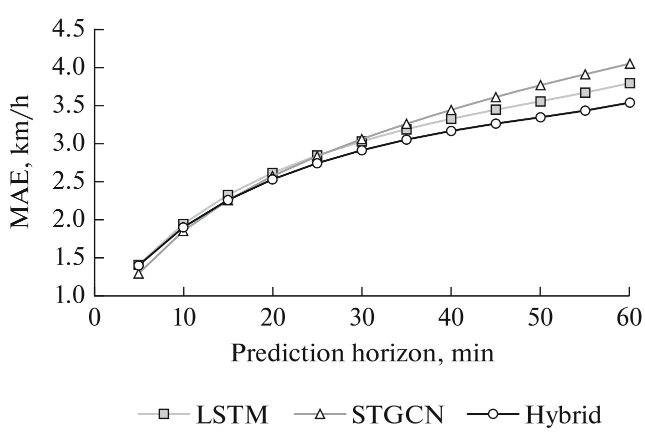

Fig. 3. Comparison of prediction accuracy by MAE criteria on the PeMSD7 dataset.

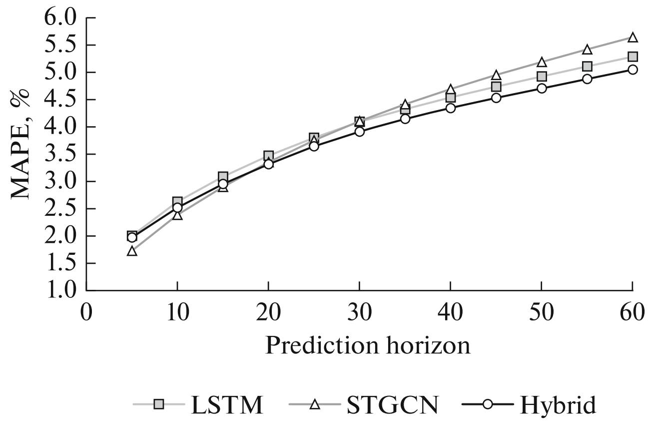

Fig. 4. Comparison of prediction accuracy by MAPE criteria on the PeMSD4 dataset.

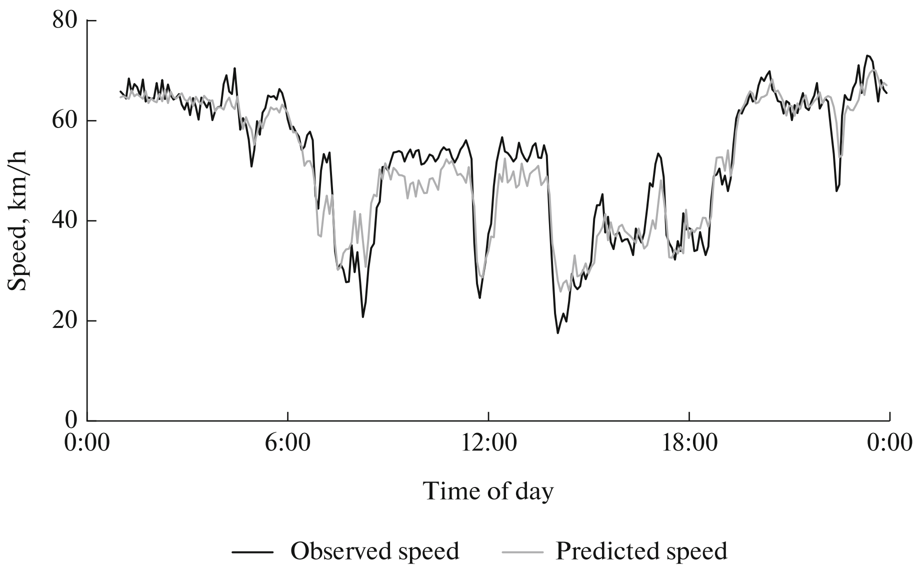

Fig. 5. Speed prediction in morning and evening rush hours for 30 min ahead.

in traffic flow. Experiments conducted on two real-world datasets show that the proposed model provides traffic data predictions with high accuracy in real-time.

In future studies, we aim to take into account other factors such as weather conditions, public events, or traffic accidents. We also plan to study more complicated model architectures, for example, with the attention mechanism.

Table 3. Computation time of training and forecasting

Model	PEMSD7		PEMSD4	
	training time, s	testing time, ms	training time, s	testing time, ms
LSTM	11.5	5.32	16.2	5.25
STGCN	116.08	33.05	440.59	43.08
ASTGCN	32.11	46.43	120.45	59.98
Hybrid	12.78	13.63	25.7	20.47

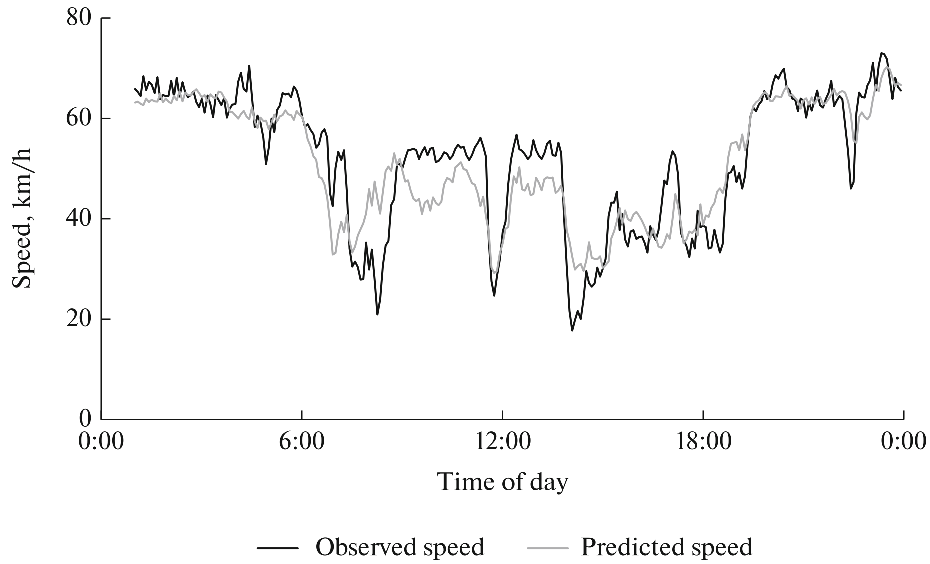

Fig. 6. Speed prediction in morning and evening rush hours for 60 min ahead.

FUNDING

The work was partially supported by Russian Foundation for Basic Research research projects nos. 18-07-00605 A, 18-29-03135-mk.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Qureshi, K.N. and Abdullah, A.H., A survey on intelligent transportation systems, *Middle East J. Sci. Res.,* 2013, vol. 15, no. 5, pp. 629–642. https://doi.org/10.5829/idosi.mejsr.2013.15.5.11215
2. Patel, P., Narmawala, Z., and Thakkar, A., A survey on intelligent transportation system Using internet of things, *Adv. Intell. Syst. Comput.,* 2019, vol. 882, pp. 231–240. https://doi.org/10.1007/978-981-13-5953-8_20
3. Agafonov, A.A. and Yumaganov, A.S., Bus arrival time prediction using recurrent Neural Network with LSTM architecture, *Opt. Mem. Neural Networks,* 2019, vol. 28, no. 3, pp. 222–230. https://doi.org/10.3103/S1060992X19030081
4. Agafonov, A. and Myasnikov, V., Stochastic on-time arrival problem with levy stable distributions, 2019, pp. 227–231. https://doi.org/10.1109/ICITE.2019.8880254
5. Vlahogianni, E.I., Karlaftis, M.G., and Golias, J.C., Short-term traffic forecasting: Where we are and where we're going, *Transp. Res., Part C: Emerging Technol.,* 2014, vol. 43, pp. 3–19. https://doi.org/10.1016/j.trc.2014.01.005
6. Lana, I., Del Ser, J., Velez, M., and Vlahogianni, E.I., Road traffic forecasting: recent advances and new challenges, *IEEE Intell. Transp. Syst. Mag.,* 2018, vol. 10, no. 2, pp. 93–109. https://doi.org/10.1109/MITS.2018.2806634
7. Ahmed, M.S. and Cook, A.R., Analysis of freeway traffic time-series data by using Box-Jenkins techniques, *Transp. Res. Rec.,* 1979, no. 722, pp. 1–9.
8. Williams, B.M. and Hoel, L.A., Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, *Transp. Eng. J.,* 2003, vol. 129, no. 6, pp. 664–672. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664)
9. Chandra, S.R. and Al-Deek, H., Predictions of freeway traffic speeds and volumes using vector autoregressive models, *J. Intell. Transp. Syst.: Technol., Plann. Oper.,* 2009, vol. 13, no. 2, pp. 53–72. https://doi.org/10.1080/15472450902858368

10. Sun, S., Zhang, C., and Yu, G., A Bayesian network approach to traffic flow forecasting, *IEEE Trans. Intell. Transp. Syst.,* 2006, vol. 7, no. 1, pp. 124–133.
https://doi.org/10.1109/TITS.2006.869623

11. Zheng, W., Lee, D.-H., and Shi, Q., Short-term freeway traffic flow prediction: Bayesian combined neural network approach, *Transp. Eng. J.,* 2006, vol. 132, no. 2, pp. 114–121.
https://doi.org/10.1061/(ASCE)0733-947X(2006)132:2(114)

12. Wu, C.-H., Ho, J.-M., and Lee, D.T., Travel-time prediction with support vector regression, *IEEE Trans. Intell. Transp. Syst.,* 2004, vol. 5, no. 4, pp. 276–281.
https://doi.org/10.1109/TITS.2004.837813

13. Agafonov, A.A., Yumaganov, A.S., and Myasnikov, V.V., Big data analysis in a geoinformatic problem of short-term traffic flow forecasting based on a K nearest neighbors method, *Comput. Opt.,* 2018, vol. 42, no. 6, pp. 1101–1111.
https://doi.org/10.18287/2412-6179-2018-42-6-1101-1111

14. Tian, Y., Zhang, K., Li, J., Lin, X., and Yang, B., LSTM-based traffic flow prediction with missing data, *Neurocomputing,* 2018, vol. 318, pp. 297–305.
https://doi.org/10.1016/j.neucom.2018.08.067

15. Xu, J., Rahmatizadeh, R., Boloni, L., and Turgut, D., Real-Time prediction of taxi demand using recurrent neural networks, *IEEE Trans. Intell. Transp. Syst.,* 2018, vol. 19, no. 8, pp. 2572–2581.
https://doi.org/10.1109/TITS.2017.2755684

16. Fu, R., Zhang, Z., and Li, L., Using LSTM and GRU neural network methods for traffic flow prediction, 2017, pp. 324–328.
https://doi.org/10.1109/YAC.2016.7804912

17. Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., and Liu, J., LSTM network: A deep learning approach for Short-term traffic forecast, *IET Intell. Transp. Syst.,* 2017, vol. 11, no. 2, pp. 68–75.
https://doi.org/10.1049/iet-its.2016.0208

18. Zhang, S., Tong, H., Xu, J., and Maciejewski, R., Graph convolutional networks: a comprehensive review, *Comput. Soc. Networks,* 2019, vol. 6, no. 1.
https://doi.org/10.1186/s40649-019-0069-y

19. Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y., Spectral networks and deep locally connected networks on graphs, *2nd Int. Conf. on Learning Representations, ICLR 2014—Conference Track Proceedings, 2014.*

20. Gao, H., Wang, Z., and Ji, S., *Large-Scale Learnable Graph Convolutional Networks,* 2018, pp. 1416–1424.
https://doi.org/10.1145/3219819.3219947

21. Kipf, T.N. and Welling, M., Semi-supervised classification with graph convolutional networks, *5th Int. Conf. on Learning Representations, ICLR 2017 – Conference Track Proceedings, 2019.*

22. Yu, B., Yin, H., and Zhu, Z., Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2018, vol. 2018-July, pp. 3634–3640.

23. Cui, Z., Henrickson, K., Ke, R., and Wang, Y., Traffic graph convolutional recurrent Neural Network: A deep learning framework for network-scale traffic learning and forecasting, *IEEE Trans. Intell. Transp. Syst.,* 2019, pp. 1–12.
https://doi.org/10.1109/TITS.2019.2950416

24. Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H., *Attention based spatial-temporal graph convolutional networks for traffic flow forecasting*, *AAAI,* 2019, vol. 33, pp. 922–929.
https://doi.org/10.1609/aaai.v33i01.3301922

25. Guo, G. and Yuan, W., Short-term traffic speed forecasting based on graph attention temporal convolutional networks, *Neurocomputing,* 2020, vol. 410, pp. 387–393.
https://doi.org/10.1016/j.neucom.2020.06.001

26. Lu, Z., Lv, W., Cao, Y., Xie, Z., Peng, H., and Du, B., LSTM variants meet graph neural networks for road speed prediction, *Neurocomputing,* 2020, vol. 400, pp. 34–45.
https://doi.org/10.1016/j.neucom.2020.03.031

27. Hammond, D.K., Vandergheynst, P., and Gribonval, R., Wavelets on graphs via spectral graph theory, *Appl. Comput. Harmonic Anal.,* 2011, vol. 30, no. 2, pp. 129–150.
https://doi.org/10.1016/j.acha.2010.04.005

28. Hochreiter, S. and Schmidhuber, J., Long short-term memory, *Neural Comput.,* 1997, vol. 9, no. 8, pp. 1735–1780.
https://doi.org/10.1162/neco.1997.9.8.1735

29. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z., Freeway performance measurement system: Mining loop detector data, *Transp. Res. Rec.,* 2001, no. 1748, pp. 96–102.
https://doi.org/10.3141/1748-12

30. Chollet, F., *Keras,* 2015. https://keras.io.