Neural Networks 145 (2022) 233–247

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction☆

Ahmad Ali [a], Yanmin Zhu [a,*], Muhammad Zakarya [b,*]

[a] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[b] Department of Computer Science, Abdul Wali Khan University, Pakistan

ARTICLE INFO

Article history:
Received 17 May 2021
Received in revised form 20 October 2021
Accepted 22 October 2021
Available online 28 October 2021

Keywords:
Traffic flow prediction
Spatial and temporal dependencies
GCN
LSTM
Road safety

ABSTRACT

The prediction of crowd flows is an important urban computing issue whose purpose is to predict the future number of incoming and outgoing people in regions. Measuring the complicated spatial–temporal dependencies with external factors, such as weather conditions and surrounding point-of-interest (POI) distribution is the most difficult aspect of predicting crowd flows movement. To overcome the above issue, this paper advises a unified dynamic deep spatio-temporal neural network model based on convolutional neural networks and long short-term memory, termed as (DHSTNet) to simultaneously predict crowd flows in every region of a city. The DHSTNet model is made up of four separate components: a recent, daily, weekly, and an external branch component. Our proposed approach simultaneously assigns various weights to different branches and integrates the four properties' outputs to generate final predictions. Moreover, to verify the generalization and scalability of the proposed model, we apply a Graph Convolutional Network (GCN) based on Long Short Term Memory (LSTM) with the previously published model, termed as GCN-DHSTNet; to capture the spatial patterns and short-term temporal features; and to illustrate its exceptional accomplishment in predicting the traffic crowd flows. The GCN-DHSTNet model not only depicts the spatio-temporal dependencies but also reveals the influence of different time granularity, which are recent, daily, weekly periodicity and external properties, respectively. Finally, a fully connected neural network is utilized to fuse the spatio-temporal features and external properties together. Using two different real-world traffic datasets, our evaluation suggests that the proposed GCN-DHSTNet method is approximately 7.9%–27.2% and 11.2%–11.9% better than the AAtt-DHSTNet method in terms of RMSE and MAPE metrics, respectively. Furthermore, AAtt-DHSTNet outperforms other state-of-the-art methods.

1. Introduction

Traffic flow prediction, as one of the most critical issues in today's smart city study, has received significant attention in the field of artificial intelligence (AI) research. With a growing number of population, accurate flow prediction (e.g., traffic flow, crowd flow) becomes increasingly crucial for first-tier cities (Helbing et al., 2015). In practice, higher prediction accuracies in traffic flow and crowd flow prediction boost the performance of many applications such as dynamic traffic management and intelligent service allocation (Wu & Tan, 2016). On the other side, a larger amount of available data has been driving the AI flow prediction research as well. Traffic flow is one of the most important task in urban traffic forecast and management. Subsequently, they are affective in reducing the congestion, re-routing the traffic to reduce traffic jam, accidents, and most importantly avoiding the crowd or flow. Furthermore, this can bring significant benefits to humans and their lives. Modern AI and learning methods such as machine learning, deep neural network, can be used to explore various aspects in the intelligent transportation system (ITS) research.

Particularly, flow refers to the number of vehicles or people who enter the city is (inflow) and departing from the city is (outflow) — as shown in Fig. 1. The objective of the crowd flow prediction problem is to predict the future times flow in real time based on historical data derived from spatio-temporal patterns. Prior to the advent of deep learning, flow prediction relied mainly on technologies developed by the time series analysis community. Traditional statistical models for flow prediction include

☆ This paper is a substantially improved version of our previously published article (Ali et al., 2019) that was demonstrated in the 25th ICPADS i.e. International Conference on Parallel and Distributed Systems, IEEE, 2019; and is published online in proceedings of the IEEE conference.

* Corresponding authors.
E-mail addresses: ahmadali@sjtu.edu.cn (A. Ali), yzhu@sjtu.edu.cn (Y. Zhu), mohd.zakarya@awkum.edu.pk (M. Zakarya).

https://doi.org/10.1016/j.neunet.2021.10.021
0893-6080/

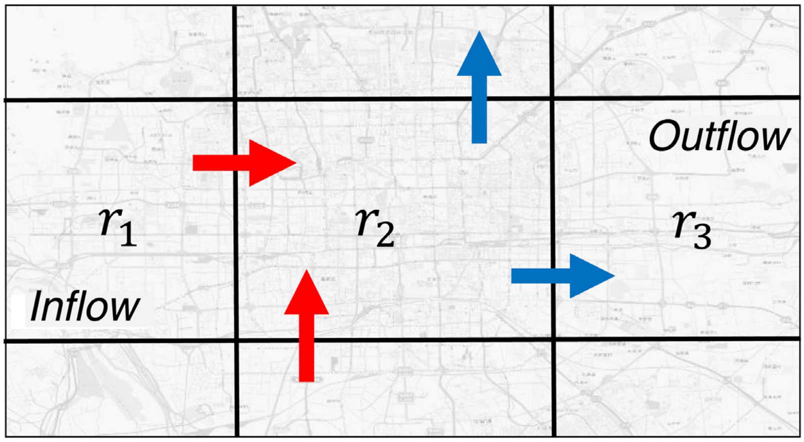

Fig. 1. Crowd flows in a specific region (Ali, Zhu, Chen, Yu, & Cai, 2019).

the Kalmen filtering, Auto-Regressive Integrated Moving Average (ARIMA), and Vector Auto-Regressive (VAR) models (Chandra & Al-Deek, 2009; Li et al., 2012; Moreira-Matias, Gama, Ferreira, Mendes-Moreira, & Damas, 2013). Although, they are easy to deploy and straight-forward, but these traditional models are unable to measure the sophisticated spatial dependencies and put a limit on their performance. We are that in some research areas, for example the prediction of the dynamic workload in clouds, these model might not be appropriate.

Unfortunately, predicting dynamically both inflow and outflow in each area is a demanding task and has remained largely unexplored in the existing literature. The traffic crowd flows is a spatio-temporal graph problem. In each region, the following major complex factors will fundamentally affect both inflows and outflows: **(i) Spatial dependencies:** Inflow of r_2 region is influenced through outflow of the distant and nearby regions i.e., r_1 and r_3 – as shown in Fig. 1. Likewise, the outflow of r_2 can influence the inflow of near regions. Besides, the inflow of r_2 region might influence its own outflow. **(ii) Temporal dependencies:** Traffic congestion changes slowly in consecutive time and show periodic patterns. The traffic jam at 8am, for instance, would certainly affect the traffic at 9am. The peak hours in the afternoon and morning on weekdays may have more similar patterns (Cheng, Tao, Zhan, Li, & Li, 2020b).

In recent years, a significant achievement in urban traffic crowd flow prediction has been achieved based on deep learning methods with high-dimensional spatio-temporal data (Xu, Wang, Long, Wang, & KLiss, 2018; Zhang, Zheng, & Qi, 2017; Zhang, Zheng, Qi, Li, & Yi, 2016; Zhang, Zheng, Sun, & Qi, 2020c). In all these works, a city is divided into a grid map based on longitude and latitude, as shown in Fig. 1, respectively (Duan et al., 2019). In Zhang et al. (2017), a convolutional neural network (CNN) approach is proposed to provide better results than the existing approaches based on neural networks. On the one hand, the ST-ResNet technique collects factors from surrounding regions through a residual-based network for spatial dependency. On the other hand, this type of data is not very good for forecasting the short-term traffic flows. Moreover, this method looks for categorization of the temporal dependency, manually; which is really difficult to express very well for the temporal dependency. This model is unable to depict the strength of various dependency of different time span. Thus, the temporal correlations overcome the factors coming from areas that are enormously faraway from the target regions. However, still these approaches fail to collectively describe the spatio-temporal correlations of crowd flows (Helbing et al., 2015).

The motivation behind using the hybrid GCN-DHSTNet model is to effectively and accurately predict the short-term traffic crowd flow in regions that could help urban managers to improve traffic management, road safety, and efficiency. Traffic flow prediction enables a variety of intelligent applications. It can assist citizens with travel planning and departure times. Furthermore, the hybrid GCN-DHSTNet model opens up new possibilities for artificial intelligence (AI) techniques in the creation of ITS, which is advantageous for the development of smart cities in the new era. The construction of urban ITS relies heavily on traffic flow prediction. Predicting traffic flow statistics accurately and rapidly is a difficult undertaking and a tedious activity. As previously said, traffic flow has a broad scale and complicated characteristics, and it is significantly influenced by the temporal and geographical factors, as well as, the external environment and road conditions. The spatio-temporal features have been employed in a variety of studies to forecast traffic flow. However, if only temporal and geographical dependencies are taken into account, then the prediction accuracy may be low, and the deviation may be substantial in some unique conditions. This is due to the facts that largely the traffic flow prediction is influenced by a variety of factors, both internal and external. Internal factors are things like road features, vehicle operating, and other things that are difficult to forecast precisely. Changes in weather, large gatherings, holidays, traffic accidents, and the like are examples of external elements that have a predictable pattern and may be forecasted accurately to some extent (Ali, Zhu, & Zakarya, 2021b).

The uncertainty induced by numerous external environmental elements has a bigger impact on traffic flow forecast, and irregular traffic flow fluctuations will potentially affect the prediction's accuracy. The existing state-of-the-art studies are largely focused on forecasting the crowd flow in a regular grid region. However, the city is normally separated by road network, which is very irregular. Forecasting the crowd flow in such regions is of great importance to public safety, transportation management, and traffic control decisions. The citywide traffic crowd flows prediction is a spatio-temporal graph problem. However, due to spatial correlation and interaction between different regions, it is hard and difficult to predict the traffic crowd flows; and they are affected by some key factors such as: (i) multiple time correlations among various time intervals (recent, daily, and weekly); (ii) complex external factors (events, weather); and (iii) meta-features (day-time, weekday/weekend), as described in Ali et al. (2021b) and Cheng, Lubamba, and Liu (2020a).

To overcome the above mentioned issues, in this paper, we propose a novel spatio-temporal neural network to simultaneously predict the traffic flows at every region on the traffic network. It is a **d**eep **h**ybrid **s**patio-**t**emporal **d**ynamic **net**works, called DHSTNet. The proposed approach forecasts both the inflow and outflow of the regions at the same time. We further develop and integrate Graph Convolutional Network (GCN) with our previously proposed DHSTNet model, namely (GCN-DHSTNet) – which is capable to concurrently capture all dependencies; and thus are more useful as the spatio-temporal dependencies can interrelate to each other. The DHSTNet model employees GCN to characterize traffic flows between nodes, thus able to capture the global spatial relationships of traffic flow between both adjacent and non-adjacent traffic nodes. This feature enables DHSTNet to work even when adjacent nodes compared to traditional CNN network. To the best of our knowledge, we believe that the proposed GCN-DHSTNet method is the first work that can synchronously measure both types of dependencies. The main contributions of this paper are highlighted as follows:

- the spatio-temporal features of citywide traffic crowd flows data are dynamically learned by our proposed DHSTNet model;
- we design DHSTNet, an approach that takes into account the spatio-temporal dependencies, as well as, other external factors such as road condition. We categorize the temporal characteristics of the citywide traffic crowds into three main sections, comprising a recent, a daily, and a weekly component;
- we further design Graph Convolutional Network with DHSTNet, called as GCN-DHSTNet, which can simultaneously handle sophisticated and dynamic spatio-temporal dependencies of urban traffic flows. We introduce an enhanced GCN model in order to learn the spatial dependency of the dynamic traffic flows. Then, in the time-series of traffic status, we apply the LSTM model to capture the dynamic temporal correlations.
- extensive evaluations on two real-world datasets by using NYC bike data and Taxi Beijing data. The extensive simulation results indicate that, the notable improvement of GCN-DHSTNet over the existing methods on various metrics.

The earlier version of this paper was presented in an international conference and published in the IEEE conference proceedings (Ali et al., 2019). In this current work, we design Graph Convolutional Network with our already existing DHSTNet model, termed as GCN-DHSTNet, to collectively predict spatio-temporal correlations of the traffic flows. Furthermore, we perform more in-depth quantitative studies and equate them to more current state-of-the-art models in a variety of settings such as: (i) impact of various configurations comprising the effect of network depth; and (ii) performance of four components.

The aforementioned ConvLSTM method has two main flaws that can be addressed with a GCN module: (i) it forgets prior outputs (from various layers); and (ii) over-fitting. Actually, GCN is a type of CNN, while LSTM is a type of RNN. Largely, CNN is employed to capture spatial dependence; however, RNN is employed to capture time dependence. Since, traffic prediction uses both dependencies to get better performance, therefore we use GCN and LSTM together. The traffic of local area is not only correlated with its nearest neighbours but also influenced by other factors. Furthermore, the fundamental benefit of GCN is its computational efficiency. The biggest feature of CNN is the requirement of locally correlated. However, the GCN framework has no such kind of limitations. Note that, GCN is a type of neural network that deals with the graph data structure. The GCN model is thought to be based on the idea of aggregating node information utilizing edge information to produce new node representations. The essence of GCN is employed to extract the topological graph's spatial properties. The node and structural information must be considered while analysing the graph data. As a result, the GCN model can learn not only the features of the nodes but also the associated information between various nodes automatically.

We structure the rest of our paper as follows. We offer an overview of the related work in Section 2. The citywide traffic flows prediction problem is mathematically formulated in Section 3. A brief discussion to previous work and DHSTNet model is also provided to make the paper self-explanatory. In Section 4, we extend the DHSTNet model with the GCN approach i.e. "GCN-DHSTNet". Extensive simulations, experiments, and comparisons, using real-world traffic datasets, are discussed in Section 5. In Section 6, we generalize the outcomes of our proposed model using different datasets and experimental parameters. Finally, we conclude this paper in Section 7; and provide outstanding insights for future research and investigation.

2. Related work

For scientific traffic direction and the implementation of intelligent transportation system (ITS), short-term traffic flow prediction is extremely important. Currently, there are now many related studies on crowd flow prediction, as well as a variety of prediction models that can be classified into three categories: statistical, machine learning, and deep learning models.

2.1. Crowd flow prediction

Forecasting the traffic crowd flow can be seen as kind of time series prediction problem. There are a number of traditional linear models that can be used to solve this problem. The historical average model is a portable model that simply utilizes the average value of historical time series to forecast future values. The classical statistical methods used in traffic flows prediction, which includes Markov chain Abadi, Rajabioun, Ioannou, et al. (2015), HA, VAR (Zivot & Wang, 2006), Bayesian networks (Sun, Zhang, & Yu, 2006), ARIMA (Williams & Hoel, 2003) and Kalman filter (Lippi, Bertini, & Frasconi, 2013) etc. These models need similar rules, and unfortunately traffic data is more hard to meet the assumptions in reality, therefore, they usually perform poorly. Some deep learning methods, for e.g., SVM (Jeong, Byon, Castro-Neto, & Easa, 2013) and KNN (Van Lint & Van Hinsbergen, 2012) are, therefore, used to depict more complex data, however feature engineering is the main requirement. As a result, deep learning has significantly improved efficiency in a variety of domains, including image processing and speech recognition. In Chen, Chen, and Qian (2014), the authors proposed a Markov random method to classify traffic congestion areas in order to address the low resolution and ambiguity of geographical areas. In Das and Ghosh (2019), a fuzzy Bayesian method is proposed for urban traffic flow prediction in order to extract both the spatio-temporal patterns and external factors, simultaneously.

In recent years, deep learning methods in citywide traffic crowd flows have been more popular due to the tremendous capacities of neural networks. In citywide traffic crowd flows, there are mainly used two deep learning methods, such as LSTM and CNN. The model of CNN utilized in variety of application, particularly in computer vision (Krizhevsky, Sutskever, & Hinton, 2012), while the method of LSTM applied for sequence learning (Williams & Zipser, 1989). The important issue is that, recurrent neural network is unable to provide assurance the holding of information very long. In Lv, Duan, Kang, Li, and Wang (2014), the authors provide a stacked auto encoder (SAE) method, to forecast the citywide crowd flow of different nodes. In Altché and de La Fortelle (2017), the authors proposed LSTM model, which is a significant improvement over a standard recurrent neural networks. In sequence-based learning tasks, the LSTM model achieved the best performances including machine translation (Bahdanau, Cho, & Bengio, 2015), text generation (Sutskever, Martens, & Hinton, 2011), and speech recognition (Vinyals, Ravuri, & Povey, 2012), respectively. These current methods suffer from incomplete measurements of short-term temporal dependencies and neglect the complicated dynamic correlations between spatio-temporal dependencies.

Although, we need to consider both spatio-temporal features in the urban crowd flow prediction problem, however, neither CNN nor the LSTM method can apply both spatio-temporal correlations, simultaneously. To resolve this issue, a residual network was proposed in Zhang et al. (2017), called ST-ResNet, in order to predict the citywide crowd flows. This approach outperformed better than the existing time-based-series models in terms of prediction accuracy. This is more convenient for spatial and temporal correlations, but the main problem in the ST-ResNet model is to adopt an easy combination of three (3) different period of times together with spatial correlations. This proves that the method of ST-ResNet needs manual characterization of temporal dependencies and at the same time cannot be learned from the input data; thus limiting the capability of the model to capture its flexibility and effectively measure the temporal dependencies. In Helbing et al. (2015), complexity science (comprises analytic and predictive models) is suggested as a possible solution to address the crowd problem that might be unavoidable due to certain systemic instabilities such as high densities and small variations in speed.

2.2. Graph Convolutional Networks (GCN)

Traditional convolution can extract local patterns of data efficiently, but it can only be used with regular grid data. The graph convolution is a recent development that extends traditional convolution to data from graph structures. The spatial and spectral methods are the two major types of graph convolution methods. Convolution filters are applied directly to the nodes and neighbours of a graph by the spatial methods. As a result, the essence of these approaches is to pick the node's neighbourhood. In Niepert, Ahmed, and Kutzkov (2016) they proposed a heuristic linear method for selecting every centre node's neighbourhood, which performed well in social network tasks. Graph convolutions were incorporated into human action recognition tasks in Li, Li, Zhang, and Wu (2019). Many partitioning strategies were suggested here to divide each node's neighbourhood into different subsets and ensure that the number of subsets for each node is equal. The spectral methods, in which spectral analysis considered the graph convolutions locality. The authors in Bruna, Zaremba, Szlam, and LeCun (2014) introduced a general graph convolution system based on the Graph Laplacian, and then Defferrard, Bresson, and Vandergheynst (2016) improved the approach by implementing eigenvalue decomposition using Chebyshev polynomial approximation. The authors in Yao, Tang, Wei, Zheng, Yu, and Li (2018a) proposed a gated graph convolution network for traffic prediction, but the model does not take into account the complex spatial–temporal correlations of traffic data (Bai, Yao, Wang, Li, & Zhang, 2021; Fang et al., 2021; James, 2021).

Recently, a new kind of neural network based on graph theory was proposed in Estrach, Zaremba, Szlam, and LeCun (2014), called as Graph Convolution Network (GCN). In contrast to CNN, GCN tries to analyse irregular data using graphs, i.e. GCN's input data is a set of nodes with their relationships represented as edges. Then, in various study disciplines, such as action recognition (Li et al., 2019), protein interface (Fout, 2017), and semi-supervised classification (Kipf & Welling, 2017), GCN has been successfully used to achieve advances. This is also supported by the most recent investigation of GCN application in traffic data processing (Li, Yu, Shahabi, & Liu, 2018; Yu, Yin, & Zhu, 2018). Although, GCN sheds light on flow of crowd data processing, it is still unknown what elements influence traffic forecast. In particular, in existing methods, the graph is often set and formed based on prior human experience, which is insufficient to describe a complex and large-scale traffic system. Therefore, learning graph based spatio-temporal traffic flows data is an essential idea with a data-driven method.

Inspired by the works as above mentioned, we propose a novel graph convolution network with our existing DHSTNet model, termed as GCN-DHSTNet, that integrates GCN with LSTM units to jointly acquire both spatio-temporal dependencies of urban traffic flows. Table 1 summarized the comparison between our proposed model and closely related studies. The features of the above methods are also shown in James (2021). The GCN-DHSTNet-based model we propose can capture external factors as well as spatio-temporal correlations. Furthermore, the GCN-DHSTNet has the same features as DHSTNet and ST-ResNet. Daily and weekly influence are implicitly captured by the HA method, but cannot capture the spatial correlations and external factors. Hong, Yokoya, Chanussot, Xu, and Zhu (2019a), Hong, Yokoya, Ge, Chanussot, and Zhu (2019c) and Hong, Yokoya, Chanussot, and Zhu (2019b) have used the graph learning approach for semi-supervised remote sensing and image classification. Albeit, CNNs are well-known and widely used for image classification due to their capabilities to capture spatial spectral feature representations. However, for modelling relations among samples their abilities are limited. Beyond the grid sampling limitations, GCNs have successfully shown its ability in irregular (non-grid) data analysis and representation (Hong, Gao, Yao, Zhang, Plaza, & Chanussot, 2020). Furthermore, the time complexity of the GCN is reported more than the CNN approaches, in particular, when implemented in a distributed fashion (Cheng, Zhan, & Qi, 2017).

Jiang, Li, Sun, Hu, Yun, and Liu (2021a) and Jiang, Li, Tan, Huang, Sun, and Kong (2021b), used the convolution neural network (CNN) and, then, as improved Faster-RCNN algorithm for computer vision and object detection. In the actual environment, the proposed methods have shown approximately 82% accuracy. Zhang, Liu, Feng, Liu, and Ju (2019) and Zhang, Liu, Gao, and Ju (2020b), proposed planning algorithm and methods for caging objects with a desired capture capability of the dual-arm space robots. Furthermore, the idea of caging compatibility and its associated index is suggested to plan near-optimal pre-grasping configurations of the robots. Similarly, Duan et al. (2021) used the LSTM network for multimodal fusion gesture recognition in the area of computer vision and pattern recognition. Albeit, the proposed method is based on double CCN model; however, the influence of different network structures over the proposed method is not analysed. Sun et al. (2020), studied the AI methods including SVM (support vector machine) and GRNN (generalized regression neural network) for human computer interaction (HCI) and their applications in gesture, pattern recognition (Cheng et al., 2017). Cheng et al. (2020a) used LSTM and CNN approaches for activity prediction of the humans. The proposed model accounts for temporal information. Their results illustrate that it is possible to anticipate an action through combining multiple objects. In the field of visual recognition, a person re-identification is a complex task. To solve this issue, the authors in Cheng et al. (2020b, 2017) suggested a distributed deep convolutional neural network model to generate features. Moreover, the parallel implementation, e.g., distributed attributes learning and parameter manipulation are used to make the proposed model speed up.

The comparative summary among our proposed GCN-DHSTNet model and other closest rivals and related works is shown in Table 1. We strongly believe that this information would certainly help the readers of this manuscript to quickly identify opportunities for further investigation and research.

Table 1
Summary of the related work in comparison to the proposed approach.

	Spatial		Temporal			Work in this paper
Baselines	intra	inter	recent	daily influence	weekly influence	
DHSTNet (Ali et al., 2019)	✓	✓	explicit	explicit	explicit	✓
ST-ResNet (Zhang et al., 2017)	✓	✓	explicit	explicit	explicit	✓
ARIMA (Williams & Hoel, 2003)	×	×	explicit	×	×	×
HA (Zivot & Wang, 2006)	×	×	×	implicit	implicit	×
AAtt-DHSTNet (Ali, Zhu, & Zakarya, 2021a)	✓	✓	×	explicit	explicit	×
GCN-DHSTNet	✓	✓	explicit	explicit	explicit	✓

3. Preliminaries

This section introduces some primary traffic flow notations, then, illustrating the issue of traffic flows prediction.

Definition 1 (*Regions of a City*)**.** The overall city is divided in the form of (a x b) grid map with n regions, where $n = a \times b$ based on latitude and longitude characteristics; and the grid denotes an area, as per existing research works (Ma, Dai, He, Ma, Wang, & Wang, 2017; Yao, Tang, Wei, Zheng, Yu, & Li, 2018b).

Definition 2 (*Traffic Flows*)**.** We consider S to be a trajectories set at t time interval based on previous knowledge from Yao et al. (2018a). The ath row and the bth column are referred to as area (a, b). At time t, both inflow and outflow are depicted as follows by the following two equations i.e. Eqs. (1) and (2), respectively:

$$x_t^{in,a,b} = \sum_{T_r \in S} |\{t > 1 | g_{t-1} \notin (a, b) \wedge g_t \in (a, b)\}| \tag{1}$$

$$x_t^{out,a,b} = \sum_{T_r \in S} |\{t \geq 1 | g_t \in (a, b) \wedge g_{t+1} \notin (a, b)\}| \tag{2}$$

where g_t shows the geospatial coordinate; T_r: $g_1 \to g_2 \to, \dots, \to g_{|Tr|}$ is a trajectory in S, $g_t \in (a, b)$ displays the g_t point lies inside the grid (a, b).

Definition 3. At t given time interval, the inflow and outflow of traffic flows in $(a \times b)$ regions can be represented as a three-dimensional (3D) tensor $\mathbf{X_t} \in \mathbb{R}^{a \times b \times 2}$, where $(\mathbf{X_t})_0$ represents the inflow matrix, while $(\mathbf{X_t})_1$ indicates the matrix of outflow.

Definition 4. Network of Roads G. To represent the topological structure of the road sections, we utilize an unweighted graph G = (V, E) and treat every road as a segment, where V is a road segment set, V = v_1, v_2, .., v_m, E represents set of edges, and m represents the number of road sections. Adjacency matrix A is used to demonstrate the accessibility of the road network without loss of generality. When G is an unweighted network, A is a matrix composed of 1 and 0, where 1 means that corresponding road segments are related, and 0 otherwise.

The problem of predicting spatio-temporal traffic flows can thus be viewed as an analysis of the mapping function f based on topology G of the road network and matrix Y of the function and then formulating traffic flows in the next T moments, as illustrated in Eq. (3):

$$[Y_{t+1}, \dots, Y_{t+T}] = f(G; (Y_{t-n}, \dots, Y_{t-1}, Y_t)) \tag{3}$$

where n denotes the time series length and T represents the historical time series length that is required to be predicted.

Problem Statement: Given the reasons for predicting traffic flows, indicated through T = 1,2,3, …, t-1, the foremost and key objective of the urban crowd traffic flows prediction issue is to forecast both inflow and outflow for the overall city at next t time interval. The forecasted values can, then, be used either to monitor the entire smart city (traffic) and/or take appropriate traffic routing decisions. The later might be helpful in avoiding crowd while the former one can be used for managing the entire vehicular network.

4. The proposed GCN-DHSTNet framework

In this section, we present the structure of our previously published DHSTNet approach (Ali et al., 2019). Furthermore, we design and integrate a graph convolutional network with our existing model. We call the new model as GCN-DHSTNet and its main components are shown in Fig. 2.

Fig. 2 demonstrates the architecture of the proposed GCN-DHSTNet model, which is made up of four key components: recent, daily, weekly, and historical data of external factors. As described earlier in Section 3, the entire city can be assumed and divided into an $(a \times b)$-dimensional grid map. Furthermore, the time axis is divided into three distinct time segments that are used to denote: (i) distant, (ii) recent, and (iii) near history.

The three branches of temporal feature share almost similar shapes of the neural network with the LSTM network, followed by a GCN model. Furthermore, each shape consists of several spatio-temporal blocks and a, unique, fully connected layer. The ConvLSTM module and the GCN module are present in every spatio-temporal block. We utilize, for each branch, a residual-based system to boost the efficiency of the model training process. The distant regions spatial dependencies are then extracted using the GCN model. The temporal dependencies are captured for every temporal sequence through the LSTM model. We implemented a GCN-DHSTNet approach to cope the global feature series to obtain well the temporal correlations of traffic flows at various time intervals. In the external component, we aim to manually extract certain important features from the external datasets, including holiday, wind, events, weather information, etc., and feeding them into a two layers fully connected (FC) neural network. The outputs of the three divisions are combined as $\mathbf{X_F}$ using parameter matrices, with different weights applied to the effects of various features in different areas. In addition, the $\mathbf{X_F}$ is joined with external branch output as $\mathbf{X_{ext}}$. We, then, combine external as well as spatio-temporal functions. Finally, in the flow of crowds setting, we utilize a network of fully-connected to calculate the loss function (Dai, Hu, Ge, Ning, & Liu, 2021; Yuan & Li, 2021). In the rest of this section, we describe and mathematically formulate the proposed GCN-DHSTNet model along with spatio-temporal dependencies and external factors.

4.1. Modelling spatio-temporal dependencies based on graph convolutional network

In the spatial domain, the urban traffic flow patterns of different areas have various impacts among each other and the most common impact is extremely dynamic. Here, we use a graph convolution network in order to utilize the dynamic correlations between spatial domain nodes (Zhu et al., 2020). We propose a temporal graph convolutional network model based on a graph convolution network (GCN) and long short term memory (LSTM) units to capture spatio-temporal features at the same time from

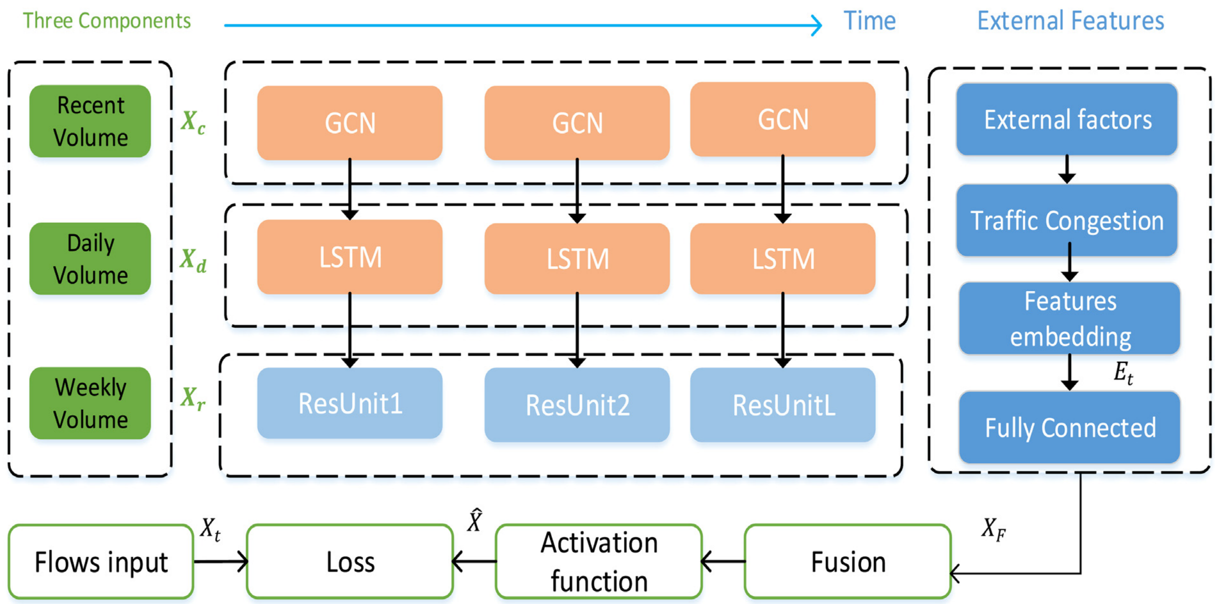

Fig. 2. Structure diagram of the proposed GCN-DHSTNet architecture, Fully connected (FC), Graph Convolutional Network (GCN), Long Short Term Memory (LSTM).

traffic flows data. At a given t time interval, a city is partitioned into an $(a \times b)$ areas, and the crowd flows can be denoted as a tensor $\mathbf{X_t} \in \mathbb{R}^{a\times b\times k}$, where k denotes the number of variables of traffic flows. The generated tensor can be measured an image of multi-channel, which includes width of pixels b, pixels k of channels, and height of pixels a. This form of image captures the spatial dependencies of the traffic crowd flows in the region (Chen, Li, Teo, Zou, Li, & Zeng, 2020; Zhang, Chang, Meng, Xiang, & Pan, 2020).

The above cases show that temporal correlations and their dependencies have a major impact on traffic circumstances, including recent, daily, and weekly, though the degree of influence varies. The first three properties, temporal recent, daily, and weekly, are chosen to be temporally dependent by using the GCN network. The output of the recent branch is formulated as $\mathbf{X}'_{c,t}$ At time t, assume a recent fragment be $\lambda_3 \circ (\lambda_1 \circ X^{L}_{c,t} + \lambda_2 \circ X^{L+1}_{c,t}) + \lambda_4 \circ X^{L+2}_{c,t}$ (this should be noted here that each element is treated as a three-dimensional tensor $\in \mathbb{R}^{2\times a\times b}$). We incorporated them by the first axis, take into account as one tensor $\mathbf{X}^{(0)}_{c,t} \in \mathbb{R}^{l_c\times 2\times a\times b}$ and followed through an LSTM layer as follows by Eq. (4):

$$\begin{aligned}
i_t =& \sigma\left(W_i X_t * \mathbf{X}^{(0)}_{c,t} + \sum_{k=0}^{k} \theta_{hki} T_k(\mathcal{C}_{t-1}) + W_{hi} * \mathbf{X}^{(1)}_{c,t-1} + b_i\right),\\
f_t =& \sigma\left(W_f X_t * \mathbf{X}^{(0)}_{c,t} + \sum_{k=0}^{k} \theta_{hkf} T_k(\mathcal{C}_{t-1}) + W_{hf} * \mathbf{X}^{(1)}_{c,t-1} + b_f\right),\\
\mathcal{C}_t =& f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\\
&\left(W_c X_t * \mathbf{X}^{(0)}_{c,t} + + \sum_{k=0}^{k} \theta_{hkc} T_k(\mathcal{C}_{t-1}) + W_{hc} * \mathbf{X}^{(1)}_{c,t-1} + b_c\right),\\
o_t =& \sigma\left(W_o X_t * \mathbf{X}^{(0)}_{c,t}\right.\\
&\left. + \sum_{k=0}^{k} \theta_{hko} T_k(\mathcal{C}_{t-1}) + W_{ho} * \mathbf{X}^{(1)}_{c,t-1} + b_o\right),\\
\mathbf{X}^{(1)}_{c,t} =& o_t \circ \tanh(\mathcal{C}_t)
\end{aligned} \tag{4}$$

where the variable X_t denotes the input of the GCN-DHSTNet model at time interval t and $\sum_{k=0}^{k} \theta_{hkf} T_k(\mathcal{C}_{t-1})$ represents the graph convolution on the hidden states. Similarly, the notation $\circ$ represents the Hadamard product.

Similarly, by using the same mathematical operations as described above, we also establish the daily, and weekly branch. We take the daily branch as an example. We determine the sequence length of three dependent components as follows: Suppose that, l_d is a daily segment time intervals and d represents daily-period. Thus, the dependent sequence of daily branch is $\lambda_3 \circ (\lambda_1 \circ X^{L}_{d,t} + \lambda_2 \circ X^{L+1}_{d,t}) + \lambda_4 \circ X^{L+2}_{d,t}$. The daily branch output is $\mathbf{X}'_{d,t}$. The weekly branch dependent sequence is formulated as $\lambda_3 \circ (\lambda_1 \circ X^{L}_{r,t} + \lambda_2 \circ X^{L+1}_{r,t}) + \lambda_4 \circ X^{L+2}_{r,t}$ and trend of weekly branch output is formulated as $\mathbf{X}'_{r,t}$, r represents the weekly component. Note that d and r are two different types of periods.
Where $\mathbf{X}'_{c,t}$, $\mathbf{X}'_{d,t}$, $\mathbf{X}'_{r,t}$ show the output of the three temporal components. The notation "$\circ$" represents the Hadamard product, while λ_1, λ_2, λ_3, λ_4 show the learnable parameters.

As we discussed above, recent, daily-volume, and trend-volume affect all the distinct regions and the degrees of influence on every area could be very different. Inspired by these research works, we propose a method to fuse the three temporal components, termed as parametric-tensor-based, and is denoted by Eq. (5):

$$\mathbf{X_F} = \mathbf{W_c} \circ \mathbf{X}'^{(L+2)}_{c,t} + \mathbf{W_d} \circ \mathbf{X}'^{(L+2)}_{d,t} + \mathbf{W_r} \circ \mathbf{X}'^{(L+2)}_{r,t} \tag{5}$$

where $\circ$ shows the Hadamard product, $\mathbf{X_F}$ denotes the fused features, $\mathbf{W_c}$, $\mathbf{W_d}$, and $\mathbf{W_r}$ represent the learnable parameters that equilibrium the degrees influenced through numerous spatio-temporal features.

4.2. External branch

The citywide traffic crowd flows can be influenced by several complex external factors such as holidays, weather conditions, and events. For example, during holidays celebrations such as Christmas and Chinese New Year, traffic crowd flows are heavier in some places compared to non-holidays, while in some places the traffic flows are opposite. Another best example is rainy weather might cause traffic to slow down due to slippery roads. Suppose E_t be a feature vector that shows these external influences during the time period that has been predicted. Table 2 shows the information in more detail.

In this work, our main focus is on events, holiday, weather conditions, and other metadata such as (weekday, day of week, and weekend). To predict the flows of crowds congestion at time t, we rapidly measure the holiday and meta-data, while the weather for future time interval t is unknown and hard to estimate. To solve this issue, we could use weather forecasting at approximate weather time interval $t-1$ or time interval t. Formally at E_t, we create two fully connected layers for each sub factors, the first of which can be seen as an embedding layer followed by an activation layer. On the other hand, the second layer is applied to map low to high level dimensions with the same shape as $\mathbf{X_t}$.

We explicitly integrate the output of external branches and temporal components directly as shown in Fig. 2. We defined the output of $\hat{\mathbf{X}}_\mathbf{t}$ external branch and multiple temporal components in Eq. (6).

$$\hat{\mathbf{X}}_\mathbf{t} = (\mathbf{X_F} + \mathbf{X_{ext}}) \tag{6}$$

To predict $\mathbf{X_t}$, our GCN-DHSTNet can be trained from the external component and first three temporal components by reducing the Mean Squared Error (MSE) output value among the projected flow and real flow matrix. In this study, we took the outflow and inflow predictions as two separated tasks, thus the loss function was created to reflect the effects of these two types of predictions. The first part of the loss function in inflow prediction over the θ matrix is denoted by Eq. (7), while the second part is denoted by Eq. (8):

$$Loss(\theta)_{in} = \|\mathbf{X_t} - \hat{\mathbf{X}}_\mathbf{t}\|_2^2 \tag{7}$$

where θ includes $\mathbf{W_c}$, $\mathbf{W_d}$ and $\mathbf{W_r}$ are some learnable parameters in the proposed GCN-DHSTNet framework.

The second part of the loss function is denoted by $Loss(\theta)_{out}$, which is referred to the outflow prediction.

$$Loss(\theta)_{out} = \|\mathbf{X_t} - \hat{\mathbf{X}}_\mathbf{t}\|_2^2 \tag{8}$$

4.3. Residual units

Even when the well-known regularization techniques and activation functions (e.g. *ReLU*) are used, it is difficult to train very deep convolutional neural networks (Ioffe & Szegedy, 2015; Krizhevsky et al., 2012; Nair & Hinton, 2010). On the other hand, capturing very massive citywide dependencies still requires a very deep network. Assume that the convolution kernel size is 3×3 and the input size is 32×32 for a specific traffic crowd flows. If we want to depict the citywide dependencies (i.e., every node in the high-level layer depends on all nodes in the input), it requires more than 15 consecutive convolutional layers. To solve this issue, we incorporate the residual learning approach into our model, which has been suggested to be effective for training ultra deep neural networks, in particular, with over 1000 layers (He, Zhang, Ren, & Sun, 2016).

4.4. Training and cross validation process of the GCN-DHSTNet model

Algorithm 1 outlines the training and cross validation process of the suggested GCN-DHSTNet model. The GCN-DHSTNet based trainable parameters are randomly initialized and optimized using back-propagation. The training and testing datasets are both fed, into Algorithm 1, along with external features and various components. The back-propagation approach utilizes the stochastic gradient method to minimize the proposed model loss function. To improve the overall performance of our proposed method, we utilize the dropout strategy approach. From steps 1 to 3, the input data is normalized and reshaped accordingly. The outer for loops are used to learn from the given parameters given the number of epochs and training dataset size. From the original sequence data, we first create the training instances (steps 6 to 9). Then, we choose an appropriate instance and put it in the learned parameters i.e. θ (steps 10 to 13). After initializing all learnable parameters (step 14), we, then use the back propagation method and Adam optimizer to train the GCN-DHSTNet model (steps 15 to 21). Moreover, we perform cross validation while assuming 10 folds to explicitly demonstrate unbiased results (steps 22 to 24).

Algorithm 1: The GCN-DHSTNet Algorithm

Input: Historical observations: $\{X_t | t \leftarrow 1, 2, 3, \ldots, n\}$;
Training dataset T_d, Testing dataset t_d, Epochs E, Batch size B ;
External features: $\{E_t | t \leftarrow 1, 2, 3, \ldots, n\}$;
Recent length, daily and weekly component are: l_c ,l_d ,l_r ;
Daily span d, Weekly span r ;
Output: GCN-DHSTNet model
1 D $\leftarrow \theta$ [θ includes learnable parameters $\mathbf{W_c}$, $\mathbf{W_d}$ and $\mathbf{W_r}$] ;
2 Normalize T_d between (0, 1) ;
3 Reshape T_d ;
4 **for** *Epoch* $\leftarrow$ *1 to E* **do**
5   **for** *Sample* $\leftarrow$ *1 to* T_d **do**
6   **for** *all accessible time interval t* **do**
7       $X_c \leftarrow [X_{t-l_c}, X_t - (l_{c-1}), \ldots, X_{t-1}]$;
8       $X_d \leftarrow [X_{t-l_d \times d}, X_{t-(l_d-1)\times d}, \ldots, X_{t-1}]$;
9       $X_r \leftarrow [X_{t-l_r \times r}, X_{t-(l_r-1)\times r}, \ldots, X_{t-1}]$;
10   **if** *($2 \le t \le n$)* **then**
11   Put a training instance (X_c, X_d, X_r, E_t) into D ;
12   **end if**
13   **end for**
14   initialize learnable parameters θ for the GCN-DHSTNet model ;
15   **do**
16   Lr $\leftarrow h$;
17   Lr $\leftarrow e^{-\lambda t}$;
18   Randomly chose batch of instances D_{batch} from D ;
19   Calculate θ by reducing the objective with $D_{batch} \in D$;
20   Train model with respect to $B \in D$ from T_d ;
21   **while** *model criteria is met*;
22   **end for**
23   Result $\leftarrow$ Predict on t_d ;
24 **end for**
25 **return** *GCN-DHSTNet model*

5. Performance evaluation and experiments

In this section, we evaluate the performance of our proposed DHSTNet model. We carried-out quantitative experiments on two different datasets, including bike NYC and Beijing taxi. Further, we design a GCN model with our existing approach, namely GCN-DHSTNet, which shows the simplification of predicting traffic flows (Ali et al., 2021b).

5.1. Experimental settings

We have implemented our model using the Keras (2.1.0) library for deep learning, on an eight cores Intel(R) Xeon(R) CPU. The system CPU platform was E5-2680 v4 @2.40GHZ and was running CUDA version 8.0, CUDNN version 8.0, and SQL server applications. The system main memory (RAM) was 256 GB and has 4 NVIDIA P100 GPU units. Furthermore, we have utilized two real-world traffic flow traces, i.e., the New York City and the Beijing datasets. These datasets are publicly available and have been largely used in the traffic flow prediction scenarios.[1,2] The major characteristics of the datasets are illustrated in Table 2 In our simulations and experiments, we tried to forecast both the inflow and outflow of the traffic flows on two different real-world workloads. Furthermore, the details of both public datasets that include weather information and trajectories flow are as follows:

- **TaxiBJ**: In this dataset, the 16 months taxicabs trajectories data are gathered from the Beijing city for four different durations i.e. from 1-07-2013 – 30-10-2013, 1-3-2014 – 30-6-2014, 1-3-2015 – 30-6-2015, and 1-11-2015 – 10-4-2016. The citywide traffic flows (inflow and outflow) are obtained from $>$ 34000 trajectories taxicab in Beijing. The last four weeks data are used as testing data, with the rest being used as training data.
- **BikeNYC**: The data of BikeNYC was collected from 2-5-2014 – 29-8-2014. This dataset contains approximately 4,392 crowd flow maps with a scale of 16 *times* 8 and a time interval of 1 h. Travel length, initial and final station IDs, and initial and end times are all included in the bike information. The last ten days of the dataset are used as testing dataset, with the rest being used as training dataset (Ali et al., 2021b).

5.2. Compared baselines

We compare our proposed GCN-DHSTNet model to 11 different baseline approaches as mentioned below.

- **HA:** Mean historical flows over the same time are used to predict both the movement of crowds and the flow of traffic congestion.
- **ARIMA** (Williams & Hoel, 2003): It is a well-known technique for predicting and comprehending time series problems.
- **LinUOTD** (Tong, Chen, Zhou, Chen, Wang, Yang, Ye, & Lv, 2017): It is a linear regression that is used to regularize spatial and temporal data.
- **XGBoost** (Chen & Guestrin, 2016): This approach is primarily used for boosting tree.
- **Multilayer Perception (MLP):** The neural network layers in this model are all fully linked.
- **ConvLSTM** (Xingjian, Chen, Wang, Yeung, Wong, & Woo, 2015): Convolutional layers are appended to long short term memory by ConvLSTM.
- **STDN** (Yao et al., 2018a): The model used an attention mechanism with neural deep hybrid network to learn spatial and temporal associations.
- **ST-ResNet** (Zhang et al., 2017): For prediction of citywide traffic flows, the ST-ResNet employs a residual neural network.
- **MST3D** (Chen et al., 2018): For prediction of citywide vehicle flow, the MST3D uses 3D CNNs to collectively learn spatio-temporal correlations.
- **DHSTNet** (Ali et al., 2019): This is the model that we proposed in a previous paper. The number of residual units in the TaxiBJ dataset is 12, while the BikeNYC dataset has four.
- **AAtt-DHSTNet** (Ali et al., 2021a): To dynamically forecast citywide traffic flows, this model proposed a data aggregation method.

5.3. Implementation details and evaluation metrics

Preprocessing: For TaxiBJ, we divide the map of the whole city into 32 × 32 grid-based regions, and each time span length fixed to 30 min. Likewise, for BikeNYC, we divide the map of whole city into 16 × 8 regions and the length period is set to an hour. These divisions are based on the assumptions as demonstrated in Zhang, Zheng, Qi, Li, Yi, and Li (2018). However, we are aware that various divisions of grid sizes will potentially produce different outcomes and prediction accuracies. The min–max normalization technique is, then, used to scale the values of traffic flows within the range of $[-1, 1]$. In our experiments and evaluation, we re-scaled the values of the prediction outcomes to the usual outputs and compared it to the ground value.

Hyper-parameter Settings: We implement the PyCharm libraries with Keras (version 2.1.0). The sequence length of three temporal components for the implementation of BikeNYC are set to $l_c = 4$, $l_d = 4$, and $l_r = 4$, respectively. Similarly, we set the lengths of the first 3 temporal properties are set to $l_c = 6$, $l_d = 4$, and $l_r = 4$ for the implementation of TaxiBJ data. If we take other values, then it will affect the results. The all convolutional layers and a fully-connected layers are initialized by Glorot and Bengio (2010). We applied the batch normalization, and set the size of mini-batch to 64. The learning rate (LR) is fixed to 0.001. To minimize the problem of over-fitting, The dropout strategy is set to 0.25 dropout rate. To optimize the proposed model, we use an Adam optimization by reducing the Euclidean loss (Kingma & Ba, 2015).

Evaluation Metrics: The Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) are used to evaluate the performance of our model. These metrics are specifically described as follows:

$$RMSE = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(\hat{x}_t - x_t)^2} \tag{9}$$

$$MAPE = \frac{1}{z}\sum_{i=1}^{z}\frac{|\hat{x}_t - x_t|}{x_t} \tag{10}$$

Where x_t and $\hat{x}_t$ represent the real map flow and the projected map flow, while for the validation the amount of samples is denoted by z. This should be noted that these two evaluation metrics are most widely used in the traffic flow prediction scenarios. Therefore, we also used them to evaluate the performance of our proposed technique. For both metrics, the smallest values are better than the largest values and vice versa.

5.4. Experiment results and analysis

The average output of our method are shown in Table 3 as compared with existing methods on two different datasets i.e., TaxiBJ and BikeNYC. Both inflow and outflow results of traffic flows are shown in Tables 4 and 5 in more detail. Each model is run ten times and the average output is recorded. From Tables 3 and 4, it is evident that our GCN-DHSTNet obtains better efficiency than existing models. It shows the usefulness of our model for clarifying the spatio-temporal associations at prediction of urban crowd flow.

[1] https://github.com/TolicWang/DeepST.
[2] https://www.kaggle.com/akkithetechie/new-york-city-bike-share-dataset.

Table 2
Datasets statistics and characteristics.

Dataset	TaxiBJ	BikeNYC
Type of data	Taxi	bike rent
Location	Beijing	New York
Time span	1-7-2013 – 30-10–2013	1-4-2014 – 30-09-2014
	1-3-2014 – 30-6–2014	–
	1-3-2015 – 30-6–2015	–
	1-11-2015 – 10-4–2016	–
Time interval	30 min	1 h
# Bikes/Taxis	34,000+	6,800+
Map size	32 × 32	16 × 8
External data (holidays)	41	20
Weather information	16 types such as sunny, cloudy, etc.	–
Temperature/C°	[−24.6, 41.0]	–

Our GCN-DHSTNet approach further minimizes the predicting errors, after integrated with the graph convolution spatio-temporal network. It proves that our model generalization to the task of traffic flow prediction is accurate. The proposed GCN-DHSTNet model consistently improves performance accuracy, especially in the prediction of short-term traffic flows. The reason why the GCN-DHSTNet model is superior in terms of short-term traffic flows over existing baselines, because the existing baselines only captures spatial or temporal dependency. However, the GCN-DHSTNet combines the advantage of the GCN and LSTM models for capturing dynamically the short-term spatio-temporal dependency. Traditional statistical time-series prediction models (such as ARIMA and HA) are significantly less effective. It reveals the flaws in methodologies that focus solely on the relationship between historical statistic values while ignoring the complex spatial–temporal dependency. We can easily see that the classical statistical approaches (e.g., HA and ARIMA) generally perform poorly because they are not able to efficiently handle complicated spatial–temporal data. The spatial correlations are captured as features in the regression models (such as XGBoost and LinUOTD). The XGBoost and LinUOTD models significantly outperform than HA and ARIMA, because XGBoost and LinUOTD include spatial correlations of the traffic flow.

These techniques, on the other hand, are still unable to derive complicating dynamic spatial dependencies and non-linear temporal dependencies. In addition, our model outperformed MLP and ST-ResNet. One of the major reasons is that the MLP approach cannot capture the linear mapping from historical data and cannot calculate the spatio-temporal dependencies adequately. Despite using deep residual networks to represent spatio-temporal dependencies, the convolutional results for ST-ResNet are linearly combined, which ignores the differences between the effects of short- and long-term temporal dependence. To capture spatial dependencies and three different time spans, the ST-ResNet approach applied three stacks of a residual neural network. Furthermore, ST-ResNet only employs 2D CNNs to learn low-level spatial features, then uses the *Tanh* function to learn the temporal dependency on the extracted spatial features, leaving low-level spatio-temporal relationships unexploited and ignored the temporal sequential dependency. Convolutional outcomes are combined indiscriminately, which ignores the normal effects of short-term and long-term temporal dependencies. Given their effectiveness in modelling temporal dependencies, the LSTM and GCN approaches outperformed the MLP and the classical time-series approaches by a significant and non-trivial margin.

By integrating both CNN and LSTM, the STDN and ConvLSTM approaches demonstrated a remarkable capability of modelling both the spatio-temporal dependencies. ConvLSTM and STDN, like ST-ResNet, ignore the temporal correlations with low-level spatial information. However, The LSTM used limits their efficiency in achieving long-term temporal dependencies. Furthermore, the ability to capture complex spatio-temporal correlations is limited by independent modelling of spatio-temporal dependencies. The GCN-DHSTNet prediction performance is more consistent with the growth of cardinality, while STDN is rapidly increasing the forecasting period. The reason behind is that, STDN uses multiple serial LSTMs, which degrade performance especially with larger datasets. The proposed GCN-DHSTNet also performs better than DHSTNet model because DHSTNet has two main flaws i.e. it forgets prior output and over-fitting which can be addressed through GCN-DHSTNet. Last but not least, GCN-DHSTNet significantly beats all competing baselines on both the BikeNYC and TaxiBJ datasets, obtaining the lowest MAPE and RMSE values. This is because GCN-DHSTNet takes into account the correlations between different types of moving objects (such as a taxi and bike), and it can learn more spatio-temporal aspects of a complicated traffic network to increase the prediction accuracy. In summary, by combining the GCN model and the LSTM approach, the hybrid GCN-DHSTNet can capture the spatio-temporal dependency better than several existing baselines (Ali et al., 2021a).

Table 3
Models comparisons on TaxiBJ and BikeNYC.

Baselines	BikeNYC		TaxiBJ	
	RMSE	MAPE	RMSE	MAPE
HA	14.91	39.90%	56.69	37.91%
ARIMA	10.01	28.42%	21.90	23.31%
XGBoost	6.39	23.19%	17.89	17.79%
MLP	7.51	25.10%	18.50	17.70%
ConvLSTM	7.60	25.69%	19.39	18.60%
LinUOTD	9.40	28.20%	21.90	20.20%
ST-ResNet	6.33	21.20%	16.90	15.40%
STDN	6.20	21.81%	16.50	15.90%
MST3D	5.81	20.39%	15.90	14.41%
DHSTNet	**4.96**	**20.10**%	**15.25**	**14.20**%
AAtt-DHSTNet	**3.93**	**18.86**%	**13.91**	**12.86**%
GCN-DHSTNet	**2.86**	**16.96**%	**12.89**	**11.56**%

5.5. Results of various GCN-DHSTNet models

We show the performance of our GCN-DHSTNet model, including network depth, and performance of four components. These models are named as GCN-DHSTNet, GCN-DHSTNet-C, GCN-DHSTNet-CD, and GCN-DHSTNet-CDR — as described later in Section 5.5.1. In the rest of this section, we discuss the impacts of various factors such as the number of residual units, the type of activation function, and other external factors (such as road condition in terms of single way, two day, and congestion) over the outcomes of the proposed model. A detailed investigation of the proposed model, using different datasets, additional parameters, factors, and generalization of our outcomes are further

Table 4
Inflow and outflow prediction results on BikeNYC.

Baselines	inflow		outflow	
	RMSE	MAPE	RMSE	MAPE
HA	14.03	38.86%	14.89	39.68%
ARIMA	9.96	28.96%	10.49	29.49%
XGBoost	6.90	22.90%	7.91	23.40%
MLP	7.30	24.60%	8.99	25.20%
ConvLSTM	7.80	25.60%	8.30	25.69%
LinUOTD	9.55	27.58%	10.11	28.74%
ST-ResNet	6.08	21.20%	6.63	22.20%
STDN	5.98	21.10%	6.60	21.89%
MST3D	5.66	20.20%	5.90	21.09%
DHSTNet	**4.78**	**19.60%**	**5.06**	**20.70%**
AAtt-DHSTNet	**4.28**	**19.57%**	**4.58**	**20.02%**
GCN-DHSTNet	**2.36**	**16.36%**	**2.96**	**16.88%**

Table 5
Inflow and outflow prediction results on TaxiBJ.

Baselines	inflow		outflow	
	RMSE	MAPE	RMSE	MAPE
HA	57.35	37.59%	57.68	39.57%
ARIMA	22.46	22.02%	22.89	22.19%
XGBoost	17.60	17.40%	18.20	17.60%
MLP	18.20	17.49%	18.30	18.20%
ConvLSTM	19.29	18.40%	19.90	18.59%
LinUOTD	21.09	20.10%	21.49	20.34%
ST-ResNet	16.69	15.10%	17.10	17.69%
STDN	16.39	15.20%	16.69	15.50%
MST3D	15.98	14.71%	16.11	14.85%
DHSTNet	**15.24**	**13.80%**	**15.50**	**14.30%**
AAtt-DHSTNet	**14.27**	**13.72%**	**14.49**	**13.49%**
GCN-DHSTNet	**12.88**	**11.08%**	**13.14**	**11.96%**

demonstrated, later, in Section 6. Moreover, we also investigate the impacts of the activation functions on our proposed deep learning model and its outcomes.

5.5.1. Performance of four components based on GCN-DHSTNet

Using different GCN-DHSTNet variants, we show the performance of four properties i.e., temporal recent, daily, weekly, and external components. The following is a list of various approaches with different factors.

- **GCN-DHSTNet:** This method combines all four factors i.e. recent, daily, weekly, and external properties.
- **GCN-DHSTNet-C:** The proposed method just captures the recent branch spatial and temporal dependence.
- **GCN-DHSTNet-CD:** The proposed method comprises only two features i.e. recent and daily properties.
- **GCN-DHSTNet-CDR:** This method consists of three temporal properties, including temporal recent, daily, and weekly, respectively.

Fig. 3 shows our GCN-DHSTNet model results with four properties on the BikeNYC dataset. Likewise, Fig. 4 shows GCN-DHSTNet results on TaxiBJ dataset. The GCN-DHSTNet-C only has a better prediction output than other approaches since it uses the recent branch. However, adding a daily and/or a weekly branch further increases the prediction performance and model accuracy. Taking into account the temporal dependencies could essentially boost the traffic flows prediction accuracy. This demonstrates that incorporating a daily feature has only a small and trivial impact on the TaxiBJ dataset. The MAPE and RMSE values can be decreased further by introducing other external branches. This shows that adding external branches has a significant impact on our model prediction accuracy and efficiency. Lastly, the GCN-DHSTNet model incorporates spatio-temporal properties and utilizes external branches to reduce the MAPE and RMSE metrics values. For example, using the BikeNYC dataset, the proposed GCN-DHSTNet method is approximately 27.2% and 11.2% better than the AAtt-DHSTNet approach in terms of RMSE and MAPE metrics, respectively. Similarly, in case of using the TaxiBJ dataset, the proposed GCN-DHSTNet method is approximately 7.9% and 11.9% better than the AAtt-DHSTNet approach in terms of RMSE and MAPE metrics, respectively.

5.5.2. Impact of the network depth

Fig. 5 presents the network depth effect for TaxiBJ dataset in more detail. At first the values of RMSE decrease as increases the number of residual units, and then the RMSE increases, showing that the larger the network, the better the performance. The training stage becomes more tough and the over-fitting probability increases when the residual units are ≥ 12.

5.5.3. Impacts of activation functions

As described, later in Section 6, we used three different types of activation functions i.e. Softmax, Sigmoid, and ReLU. We observed significant changes, which potentially might be related to the datasets, over the obtained results. For example, as shown in Fig. 7, since the XiAn Road Traffic dataset is large and accounts for two-way traffic, larger variations than the Q-traffic dataset, as shown in Fig. 6, were observed. This means that appropriate activation function should be used using the characteristics of the dataset and road conditions. Furthermore, each activation function combined with a particular training model produces different outcomes. For the GCN-DHSTNet model the overlap is small as compared to the classical LSTM and CNN approaches. This should be noted that smaller overlaps ensure that the model produces outcomes that are more close to each other i.e. deviate less from each other.

5.5.4. Impacts of external factors

The impact of external factors is relatively ignored in existing research (Hou, Chen, & Wen, 2021). The larger is the number of external factor the more time would be needed for training the model. However, the time may significantly variate for various training models. For example, in Section 6, the XiAn Road Traffic dataset offers congestion and weather details. While training the LSTM and CNN model, larger variation in RMSE and MAPE metrics were noted. These outcomes are considerably different, while having large variations with significant overlaps, from those that does not account for road conditions. As explained in Helbing et al. (2015), diverse parameters often result in traffic congestion even with the absence of road accidents. In the future, we will consider more external factors and their impacts on our observations and outcomes.

5.5.5. Model efficiency

Finally, we investigate the proposed model efficiency in terms of both prediction and training phases with response time. The corresponding results are shown in Table 6. The first thing to notice is that GCN-DHSTNet outperforms STDN and MST3D in both training and prediction phases. Our previous model i.e. DHSTNet also outperforms these tow approaches while the model suggested in this work outperforms DHSTNet model in terms of running times. The STDN method has the longest running time in both training and predicting techniques. The STDN essentially anticipates core values using local CNNs, and it trains and forecasts each region using a sliding window that spans the entire area. To predict the crowd values for the entire area, the BikeNYC dataset,

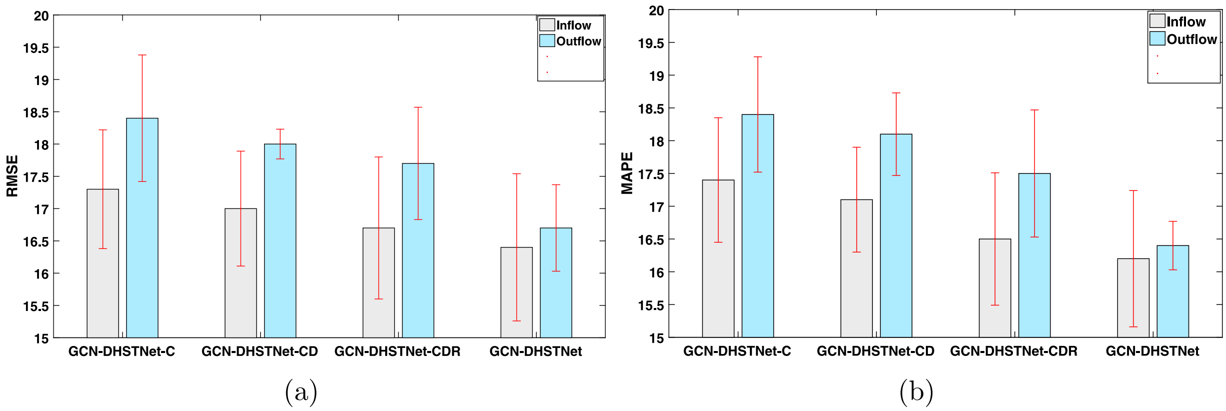

Fig. 3. Prediction results of four components on BikeNYC (a) RMSE and (b) MAPE [the error bars denote the standard deviations of various experimental runs — minimum values are best].

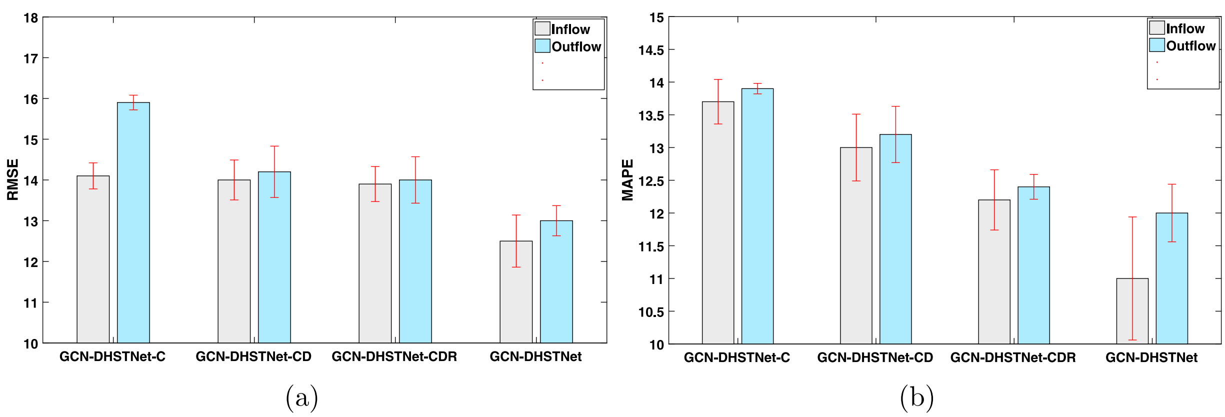

Fig. 4. Prediction results of four components on TaxiBJ (a) RMSE and (b) MAPE [the error bars denote the standard deviations of various experimental runs — minimum values are best].

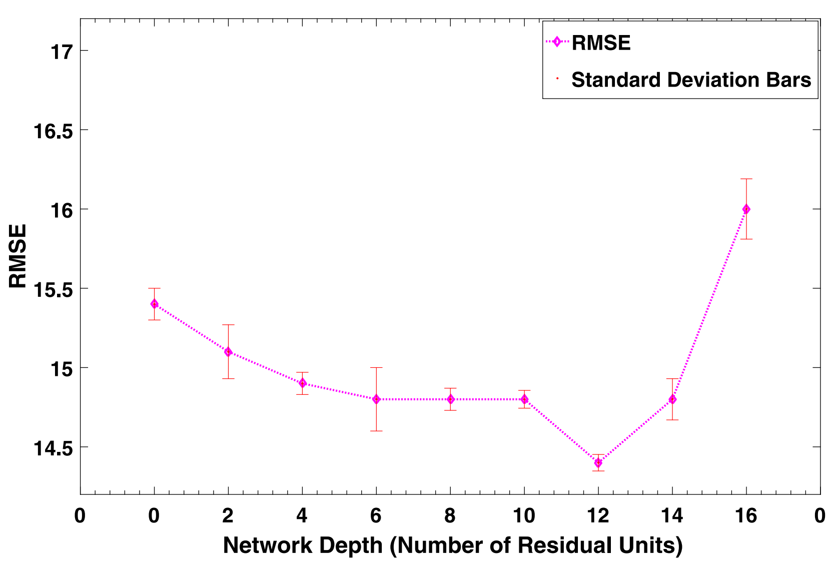

Fig. 5. Impact of the depth of the network on obtained results [the error bars denote the standard deviations of various experimental runs].

for example, must be run 8 × 16 times. On the BikeNYC dataset, the proposed GCN-DHSTNet model outperforms ST-ResNet approach by a significant margin. One of the key reasons is that the GCN-DHSTNet approach employs both GCN and LSTM, both of which are based on the graph convolutional networks, whereas ST-ResNet employs 12 residual units and two convolutional layers. Similarly, for TaxiBJ dataset, the GCN-DHSTNet outperforms both the STDN and MST3D models in training and prediction

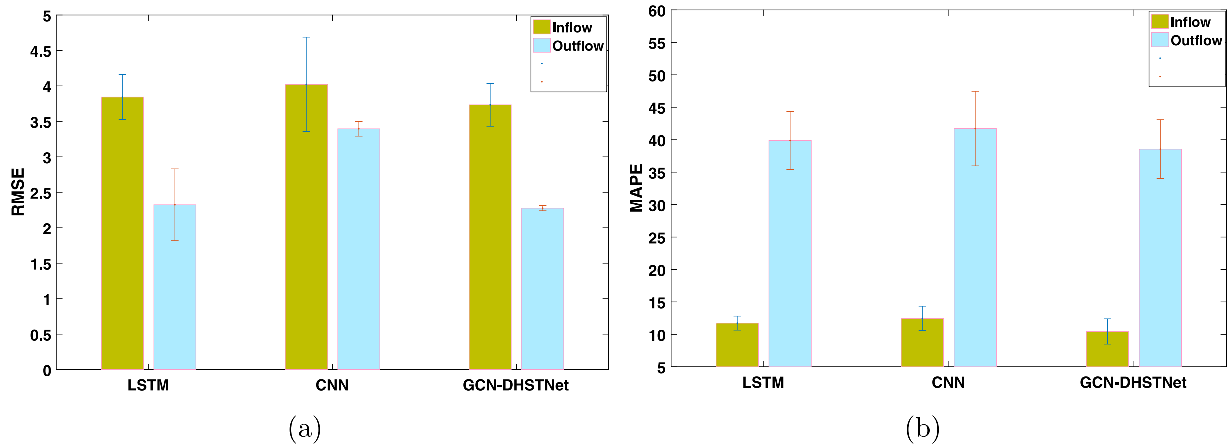

Fig. 6. Comparison results of various algorithms using Q-traffic dataset (a) RMSE and (b) MAPE [the error bars denote the standard deviations of various experimental runs — minimum values are the best].

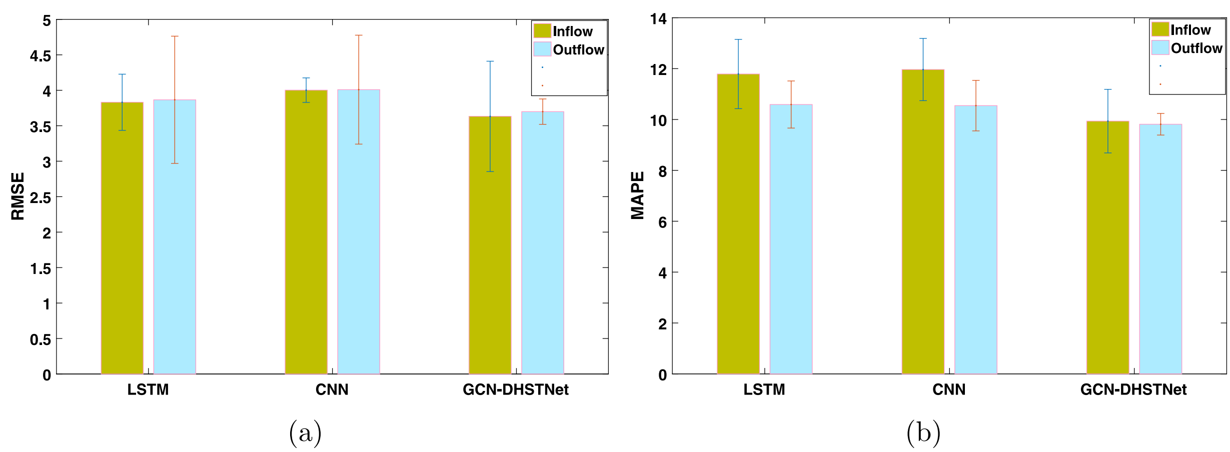

Fig. 7. Comparison results of various algorithms using XiAn Road Traffic dataset (a) RMSE and (b) MAPE [the error bars denote the standard deviations of various experimental runs — minimum values are the best].

times with significant margins. In Section 6, we generalize the outcomes of the proposed GCN-DHSTNet model using different datasets and model parameters.

6. Generalization of the outcomes

In order to find consistency in our results and scalability of our proposals, we evaluated the proposed techniques using a variety of traffic datasets and different parameters. The Q-traffic dataset is a similar trace to the TaxiBJ, collected from 01/04/2017 to 31/05/2017, but the interval time is 15 min rather than 6 min Hou et al. (2021). We also used XiAn Road Traffic dataset that has been recently collected (24-08-2019 to 24-10-2019) at 2 min time interval. Furthermore, along with two ways roads data, this trace has weekends, congestion, and weather details as well. Congestion is assumed as an additional external factor in the results presented in this section. We also used three different activation functions i.e. *Softmax*, *Sigmoid*, and *ReLU*. To better demonstrate the effectiveness of the proposed GCN-DHSTNet model, we provide a comparative study with other methods including traditional LSTM, and CNN approaches. The experiments were executed 10 times and the mean results are shown along with standard deviations. The purpose of standard deviations is to show whether there exist significant differences or the outcomes overlap with each other. In the latter case, a *t-test* should be performed to statistically validate the outcomes against other competing state-of-the-art methods.

The results, in terms of Q-traffic and XiAn Road Traffic datasets, are shown in Figs. 6 and 7, respectively. Note that variations in outcomes, as shown using error bars, are due to the activation function being used. We observed significant impact, over the outcomes, of using these activation functions. Albeit, there exist non-trivial overlaps amongst the RMSE and MAPE metrics of various models. However, our mean outcomes are consistent with our previous results. Moreover, the error bars for the GCN-DHSTNet model are significantly smaller than other techniques, with the notable exception of the inflow traffic data, as shown in Fig. 7(a), which demonstrate its efficiency. This impact is potentially due to scale of dataset and external features like congestion. This means that additional external features should be investigated (Hou et al., 2021).

As discussed earlier, the urban road network is being subdivided into various grids that both horizontally and vertically are spaced equally. If we assume that the urban network is subdivided into N horizontal parts and M vertical parts, then a total $N \times M$ rectangular grids will be the outcomes. Moreover, every grid is denoted by the graph $G(n, m)$, where $(n, m) \in Z^+$ (where Z^+ indicates a set of positive numbers). Therefore, based on the latitude and the longitude of a city, different outcomes are expected. For example, if the grid size is too small, then there is a possibility that the grid has few or no flow. This means that the grid matrix will be huge and too sparse, which may increase the evaluation complexity of the model. Duan et al. (2019) observed that smaller RMSE and MAPE values for a 24×24 grid than the

Table 6
Efficiency evaluation in terms of training and prediction response times.

Baselines	BikeNYC		TaxiBJ	
	Training time (s/epoch)	Predicting time (s)	Training time (s/epoch)	Predicting time (s)
STDN	18,973	88.7%	369,601	207.4%
MST3D	126	0.23%	5,902	2.74%
DHSTNet	125	0.21%	5,888	2.49%
GCN-DHSTNet	119	0.14%	4,852	2.31%

8×8 and the 16×16 grids. We will consider this factor in future research.

7. Conclusions and future work

In this paper, we proposed a novel deep spatio-temporal neural network model, pronounced as DHSTNet, which explicitly learns the dynamic spatio-temporal correlation of citywide traffic flows data. The proposed DHSTNet model takes into account both the LSTM and CNN models, which achieve the advantages of spatio-temporal features. We further developed a graph convolution network with our existing approach, namely GCN-DHSTNet, which simultaneously captures both the spatio-temporal characteristics along with the external branches. Using two real-world traffic datasets, we demonstrate the superiority of our proposed GCN-DHSTNet model over the prevailing state-of-the-art models. Using different real-world traffic datasets, our evaluation suggests that the proposed GCN-DHSTNet method is approximately 7.9%–27.2% and 11.2%–11.9% better than the AAtt-DHSTNet approach in terms of RMSE and MAPE metrics, respectively. Moreover, we generalized our outcomes, and offered a comparative study with the LSTM, and CNN approaches, using two other traffic datasets. Moreover, other external factors including road conditions, work, weather, and traffic congestion were also considered. Our extensive simulations show the consistent superiority of the proposed approach over other deep learning models.

In the future, we will consider other related datasets with diverse features to demonstrate the performance and accuracy of the proposed GCN-DHSTNet model. Despite the obtained results on certain parameters, there are still other essential parameters that we should consider in our future work. For example, the size of the grid division, the sequence length of the sequential components, road conditions that results in congestion, and traffic on one way and two-way highways would be other interesting points for future consideration. Furthermore, the proposed framework could be implemented over a cloud–edge-fog continuum in order to take real-time traffic routing, congestion avoidance, and migration of services (mobility) decisions through implementing the training code on a remote cloud datacentre, while the prediction code on a fog datacentre. This will potentially minimize the service latency and model training time. Similarly, the IoT devices are distributed across different regions and it is most probable that they collect duplicate and redundant data. Subsequently, redundancy increases the volume of collected data and, therefore, the model training and/or prediction time(s). In our future research, we will investigate how the redundant data can be removed to further increase the performance of the proposed model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is financially supported, in part, by the National Key Research and Development Program (No. 2018AAA0100503, No. 2018AAA0100500), National Science Foundation of China (No. 61772341, No. 61472254, No. 61772338 and No. 61672240), the Innovation and Entrepreneurship Foundation for overseas high-level talents of Shenzhen (No. KQJSCX20180329191021388), Shanghai Municipal Science and Technology Commission (No. 18511103002, No. 19510760500, 19511101500), the Program for Changjiang Young Scholars in the University of China, the Program for China Top Young Talents, the Program for Shanghai Top Young Talents, Shanghai Engineering Research Center of Digital Education Equipment, and SJTU Global Strategic Partnership Fund (2019 SJTU-HKUST).

References

Abadi, A., Rajabioun, T., Ioannou, P. A., et al. (2015). Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 653–662.

Ali, A., Zhu, Y., Chen, Q., Yu, J., & Cai, H. (2019). Leveraging spatio-temporal patterns for predicting citywide traffic crowd flows using deep hybrid neural networks. In *2019 IEEE 25th international conference on parallel and distributed systems (ICPADS)* (pp. 125–132). IEEE.

Ali, A., Zhu, Y., & Zakarya, M. (2021a). A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. *Multimedia Tools and Applications*, *80*(20), 31401–31433.

Ali, A., Zhu, Y., & Zakarya, M. (2021b). Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. *Information Sciences*, *577*, 852–870.

Altché, F., & de La Fortelle, A. (2017). An LSTM network for highway trajectory prediction. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)* (pp. 353–359). IEEE.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings*.

Bai, L., Yao, L., Wang, X., Li, C., & Zhang, X. (2021). Deep spatial–temporal sequence modeling for multi-step passenger demand prediction. *Future Generation Computer Systems*, *121*, 25–34.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. In *2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, conference track proceedings*.

Chandra, S. R., & Al-Deek, H. (2009). Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, *13*(2), 53–72.

Chen, P.-T., Chen, F., & Qian, Z. (2014). Road traffic congestion monitoring in social media with hinge-loss Markov random fields. In *2014 IEEE international conference on data mining* (pp. 80–89). IEEE.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Chen, C., Li, K., Teo, S. G., Chen, G., Zou, X., Yang, X., et al. (2018). Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction. In *2018 IEEE international conference on data mining (ICDM)* (pp. 893–898). IEEE.

Chen, C., Li, K., Teo, S. G., Zou, X., Li, K., & Zeng, Z. (2020). Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *14*(4), 1–23.

Cheng, K., Lubamba, E. K., & Liu, Q. (2020a). Action prediction based on partial video observation via context and temporal sequential network with deformable convolution. *IEEE Access*, *8*, 133527–133540.

Cheng, K., Tao, F., Zhan, Y., Li, M., & Li, K. (2020b). Hierarchical attributes learning for pedestrian re-identification via parallel stochastic gradient descent combined with momentum correction and adaptive learning rate. *Neural Computing and Applications*, *32*(10), 5695–5712.

Cheng, K., Zhan, Y., & Qi, M. (2017). AL-DDCNN: a distributed crossing semantic gap learning for person re-identification. *Concurrency Computations: Practice and Experience*, *29*(3), Article e3766.

Dai, G., Hu, X., Ge, Y., Ning, Z., & Liu, Y. (2021). Attention based simplified deep residual network for citywide crowd flows prediction. *Frontiers of Computer Science*, *15*(2), 1–12.

Das, M., & Ghosh, S. K. (2019). FB-STEP: a fuzzy Bayesian network based data-driven framework for spatio-temporal prediction of climatological time series data. *Expert Systems with Applications*, *117*, 211–227.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5-10, 2016, Barcelona, Spain* (pp. 3837–3845).

Duan, H., Sun, Y., Cheng, W., Jiang, D., Yun, J., Liu, Y., et al. (2021). Gesture recognition based on multi-modal feature weight. *Concurrency Computations: Practice and Experience*, *33*(5), Article e5991.

Duan, Z., Zhang, K., Chen, Z., Liu, Z., Tang, L., Yang, Y., et al. (2019). Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time. *IEEE Access*, *7*, 127816–127832.

Estrach, J. B., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral networks and deep locally connected networks on graphs. In *2nd international conference on learning representations, ICLR*.

Fang, S., Prinet, V., Chang, J., Werman, M., Zhang, C., Xiang, S., et al. (2021). Ms-net: Multi-source spatio-temporal network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Fout, A. M. (2017). Protein interface prediction using graph convolutional networks. (Ph.D. thesis), Colorado State University.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Helbing, D., Brockmann, D., Chadefaux, T., Donnay, K., Blanke, U., Woolley-Meza, O., et al. (2015). Saving human lives: What complexity science and information systems can contribute. *Journal of Statistical Physics*, *158*(3), 735–781.

Hong, D., Gao, L., Yao, J., Zhang, B., Plaza, A., & Chanussot, J. (2020). Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*.

Hong, D., Yokoya, N., Chanussot, J., Xu, J., & Zhu, X. X. (2019a). Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *158*, 35–49.

Hong, D., Yokoya, N., Chanussot, J., & Zhu, X. X. (2019b). Cospace: Common subspace learning from hyperspectral-multispectral correspondences. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(7), 4349–4359.

Hong, D., Yokoya, N., Ge, N., Chanussot, J., & Zhu, X. X. (2019c). Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *147*, 193–205.

Hou, Y., Chen, J., & Wen, S. (2021). The effect of the dataset on evaluating urban traffic prediction. *Alexandria Engineering Journal*, *60*(1), 597–613.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

James, J. (2021). Citywide traffic speed prediction: A geometric deep learning approach. *Knowledge-Based Systems*, *212*, Article 106592.

Jeong, Y.-S., Byon, Y.-J., Castro-Neto, M. M., & Easa, S. M. (2013). Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1700–1707.

Jiang, D., Li, G., Sun, Y., Hu, J., Yun, J., & Liu, Y. (2021a). Manipulator grabbing position detection with information fusion of color image and depth image using deep learning. *Journal of Ambient Intelligence and Humanized Computing*, 1–14.

Jiang, D., Li, G., Tan, C., Huang, L., Sun, Y., & Kong, J. (2021b). Semantic segmentation for multiscale target based on object recognition using the improved faster-RCNN model. *Future Generation Computer Systems*, *123*, 94–104.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th international conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, conference track proceedings*. OpenReview.net.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Li, B., Li, X., Zhang, Z., & Wu, F. (2019). Spatio-temporal graph routing for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33* (pp. 8561–8568).

Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., et al. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, *6*(1), 111–121.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings*. OpenReview.net.

Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 871–882.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), 818.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1393–1402.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10), June 21-24, 2010, Haifa, Israel* (pp. 807–814). Omni Press.

Niepert, M., Ahmed, M., & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In *International conference on machine learning* (pp. 2014–2023). PMLR.

Sun, Y., Xu, C., Li, G., Xu, W., Kong, J., Jiang, D., et al. (2020). Intelligent human computer interaction based on non redundant emg signal. *Alexandria Engineering Journal*, *59*(3), 1149–1157.

Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *7*(1), 124–132.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 1017–1024).

Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., et al. (2017). The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1653–1662). ACM.

Van Lint, J., & Van Hinsbergen, C. (2012). Short-term traffic and travel time prediction models. *Artificial Intelligence Applications To Critical Transportation Issues*, *22*(1), 22–41.

Vinyals, O., Ravuri, S. V., & Povey, D. (2012). Revisiting recurrent neural networks for robust ASR. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4085–4088). IEEE.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, *129*(6), 664–672.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Wu, Y., & Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802–810).

Xu, Z., Wang, Y., Long, M., Wang, J., & KLiss, M. (2018). PredCNN: Predictive learning with cascade convolutions. In *IJCAI* (pp. 2940–2947).

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018a). Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint arXiv:1803.01254.

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018b). Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint arXiv:1803.01254.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (pp. 3634–3640). ijcai.org.

Yuan, H., & Li, G. (2021). A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Science and Engineering*, *6*(1), 63–85.

Zhang, Q., Chang, J., Meng, G., Xiang, S., & Pan, C. (2020) Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34* (pp. 1177–1185).

Zhang, X., Liu, J., Feng, J., Liu, Y., & Ju, Z. (2019). Effective capture of nongraspable objects for space robots using geometric cage pairs. *IEEE/ASME Transactions on Mechatronics*, *25*(1), 95–107.

Zhang, X., Liu, J., Gao, Q., & Ju, Z. (2020b). Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. *Nonlinear Dynamics*, *100*(3), 2449–2467.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI* (pp. 1655–1661).

Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016). DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems* (p. 92). ACM.

Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., & Li, T. (2018). Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, *259*, 147–166.

Zhang, J., Zheng, Y., Sun, J., & Qi, D. (2020c). Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Transactions on Knowledge and Data Engineering*, *32*(3), 468–478.

Zhu, J., Tao, C., Deng, H., Zhao, L., Wang, P., Lin, T., et al. (2020). AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. arXiv preprint arXiv:2011.11004.

Zivot, E., & Wang, J. (2006). Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®*, 385–429.