Applied Intelligence (2023) 53:2110–2132
https://doi.org/10.1007/s10489-022-03475-7

ST-A$_G$P: Spatio-Temporal aggregator predictor model for multi-step taxi-demand prediction in cities

Manish Bhanu[1] · **Shalini Priya**[1] · **João Mendes Moreira**[2,3] · **Joydeep Chandra**[1]

Accepted: 4 March 2022 / Published online: 5 May 2022

Abstract

Taxi demand prediction in a city is a highly demanded smart city research application for better traffic strategies formulation. It is essential for the interest of the commuters and the taxi companies both to have an accurate measure of taxi demands at different regions of a city and at varying time intervals. This reduces the cost of resources, efforts and meets the customers' satisfaction at its best. Modern predictive models have shown the potency of Deep Neural Networks (DNN) in this domain over any traditional, statistical, or Tensor-Based predictive models in terms of accuracy. The recent DNN models using leading technologies like Convolution Neural Networks (CNN), Graph Convolution Networks (GCN), ConvLSTM, etc. are not able to efficiently capture the existing spatio-temporal characteristics in taxi demand time-series. The feature aggregation techniques in these models lack channeling and uniqueness causing less distinctive but overlapping feature space which results in a compromised prediction performance having high error propagation possibility. The present work introduces **Spatio-Temporal Aggregator Predictor** (**ST-A$_G$P**), a DNN model which aggregates spatio-temporal features into (1) non-redundant and (2) highly distinctive feature space and in turn helps (3) reduce noise propagation for a high performing multi-step predictive model. The proposed model integrates the effective feature engineering techniques of machine learning approach with the non-linear capability of a DNN model. Consequently, the proposed model is able to use only the informative features responsible for the objective task with reduce noise propagation. Unlike, existing DNN models, **ST-A$_G$P** is able to induce these qualities of feature aggregation without the use of Multi-Task Learning (MTL) approach or any additional supervised attention that existing models need for their notable performance. A considerable high-performance gain of 25 − 37% on two real-world city taxi datasets by **ST-A$_G$P** over the state-of-art models on standard benchmark metrics establishes the efficacy of the proposed model over the existing ones.

Keywords Spatio-temporal · Prediction · Taxi-demand · Origin-destination tensor

1 Introduction

On-demand taxi services such as Uber, Ola, and Mobike are playing a major role in meeting the transportation demands in several urban locations. Modeling the taxi demand across different regions of a city can be useful in determining the business strategies for these transport companies. Taxi-Demand prediction is among the utmost tasks from the point of business perspective as well as city traffic policy makers. Researchers have been working hard to create more efficient models to have an accurate prediction of taxi-demands at a region over time. Such models can be helpful in maintaining customer satisfaction by providing hassle-free services with manageable resources [1–4]. The following paragraphs detail the taxi-demand prediction problem and the existing models in the field of taxi-demand prediction along with their contributions and limitations

✉ Manish Bhanu
manish.pcs16@iitp.ac.in

Shalini Priya
shalini.pcs16@iitp.ac.in

João Mendes Moreira
jmoreira@fe.up.pt

Joydeep Chandra
joydeep@iitp.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, Bihar, India

[2] Department of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal

[3] LIAAD, INESC TEC, Porto, Portugal

which progress to summarize the need for an improved predictive model in the domain.

Many predictive models have been developed to assist these taxi companies as well as commuters for accurate on-demand services as possible [5–8]. To model every region of a complete city, these models primarily take as input Origin-Destination Tensor (ODT). A general form of an ODT is a three-dimensional data-structure with time (t), source (s) and destination (d) being its three dimensions. A value in a particular cell $< t_i, s_j, d_k >$ is the demand of taxi from source (s_j) to destination (d_k) in a particular time duration (t_i). The objectives of these predictive models are both to predict demands at each region and demands across each region pair over one or more steps in the future. Demand at a region is the aggregated values of all demands whose destination is that particular region irrespective of source regions. While demands between regions mean the demands between two regions when one is treated as a source and the other as destination and vice-versa. However, the existing models [9–11] target more on the region demand prediction. In this paper, we consider the demand across each region pair since it contains more information than the former one. For this purpose, researchers have developed many efficient classical models as well as statistical and Tensor-Decomposition based models. Recently advances in Deep Neural Networks, have provided many predictive models that yield ever-high accuracy in this application domain. However, the existing models have many pitfalls that create gaps between efficient feature extraction and predictive performance. Consequently, models based on different approaches could not cope with the benefits of others e.g. machine learning based models could not compete with the high accuracy and non-linearity of Deep Neural Network models. Despite all, the contribution of the existing models in the field of taxi-demand prediction has reached a considerable height considering their improvement over their competitive counterparts. These improvements can be observed precisely with the advent of a new prediction model in the domain.

The classical parametric and statistical models like ARIMA [12], SARIMA [13], VAR [14], Bayesian Update [11] etc. are able to predict the taxi-demand time-series but their performance have limited gain. Additionally, these linear models are not much helpful in predicting taxi-demand with the complex spatio-temporal features. Though, these models are able to make multi-step prediction for many regions, dealing with every region altogether along with capturing inter-regional correlations is often neglected, resulting in a marginal predictive performance gain of these models. Additionally, these models have hardly any provision to include region-specific traffic characteristics. Consequently, these models are not able to fully capture spatio-temporal characteristics of the taxi-demand time-series of a complete city.

Tensor decomposition based models are able to adapt to multi-dimensional traffic features and require no training process for the prediction. These tensor based data-driven approaches such as DTC [11], CP-Decomposition [15] and TeDCAN [16] are a few of the models in this category that have been applied for traffic time-series. These models are based on spectral decomposition of higher order tensor (ODT) and hence are able to capture spatio-temporal characteristics in-efficiently. Though most of the Tensor-Decomposition based models do not incorporate network specific-characteristics in **traffic time-series prediction**, few models like TeDCaN [16] can include specific characteristics of a 'city traffic network' for better prediction results. Despite the considerable performance of these models, they are not much efficient in long-term prediction and additionally these models being data-driven approaches are not able to fully capture complex spatio-temporal characteristics of the city taxi-demand time-series present in latent space. Though, these models require no training process, during prediction these models involve computationally costly tensor decomposition operation using approximation and require convergence by alternating least square (ALS) or a similar approach using many iterations, which consumes time. These models despite showing good short-term prediction performance often are not preferred for long term prediction due to hard to accommodate attention mechanism which recent deep neural networks models can easily integrate.

Recently, Deep Neural Networks have shown some remarkable accuracy in predicting taxi-demand time series of regions in a city. DNN layers like Convolution Neural Networks (CNN), Graph Convolution Networks (GCN), Long Short-Term Memory RNN (LSTM), convLSTM, etc. are specialized in capturing spatial and temporal characteristics in input datasets. Hence trendy predictive models like GEML [17], STMGCN [9], ST-GCN [18], att-ConvLSTM [19], T-GCN [20] etc. using these layers or their modification have proved to be an efficient predictive model with high accuracy on many taxi-demand datasets. Att-ConvLSTM [19] uses convolution operation on city grid structure (city ODT) for spatial and temporal features encoding and decoding. To enhance the prediction result, att-ConvLSTM [19] uses temporal representatives of city grid structures using kmeans++ as attention to the derived spatio-temporal encoded features by the model. The use of convolution (CNN) in att-ConvLSTM limits its capability to consider only the neighboring regions for consideration while any distant region like (airport, city-centre) are important irrespective of their distance from the region. To overcome this limitation CNN is replaced with GCN. The models like ST-GCN, STMGCN, GEML [9, 17, 18] etc. use GCN for spatial convolution. ST-GCN [18] uses spectral graph convolution along with GLU for traffic time-series

prediction. This provides a generic approach to capture spatio-temporal characteristics of the city traffic demand time-series prediction. ST-GCN performance can not compete with more recent DNN models which include different attention to incorporate traffic specific characteristics during prediction. GEML [17] is an efficient Multi-Task Learning model using GCN and LSTM. It employs traffic features for attention. Despite this, GEML's capability to capture complex latent spatio-temporal characteristics, it does not support feature refinement. Features refined for the objective task can greatly boost the performance of the model. It's an usual approach often seen for a machine learning model. Overlapping features from different attention as well as noise propagation are possible limitations [21, 22] of its predictive performance. STMGCN uses temporal attention and multiple traffic network graphs to enhance its prediction capability using GCN and RNN. STMGCN, GEML, and other trendy DNN models lack feature organization. The spatio-temporal features captured by these models have less distinctive and overlapping characteristics, which limits their predictive power. Additionally, these models have no dedicated components to reduce noise propagation during prediction. To describe, these glitches can be explained as a major gap in the utilization of input features and predictive performance. The present DNN models are less likely to utilize the conceptual benefits of feature manipulation used in the machine learning approaches that can lead to a major performance gain rather than depending on DNN models as a black-box approach.

With a view to the existing constraints in trendy predictive models, we introduce a DNN model, Spatio-Temporal Aggregator Predictor (ST-A_GP) which efficiently makes a multi-step prediction of the taxi demands among every region pair in the city. The proposed model incorporates features attention and unsupervised loss to capture complex spatio-temporal characteristics of the network. A clear advantage of ST-A_GP over other existing DNN models is that it includes a dedicated component to create non-overlapping, distinctive features from various sources and helps reduce noise propagation during prediction. Hence, the proposed model is a DNN based model that inherits the capability of feature engineering of a machine learning approach (PCA in our case) amalgamated with the highly accurate prediction capability of DNN models.. In the coming sections, we elaborate on the conceptual and constructional background of the proposed model with experimental findings to support its predictive capabilities. The major contribution of the present work can be enumerated in brief as follows:

- The paper contributes by providing a deep neural network analog of a machine learning approach (PCA) which inherits important capabilities of both approaches like high accuracy of a DNN model and efficient feature transformation of a PCA. Consequently, this has considerably improved the predictive performance of the proposed model over the existing state-of-the-art models.
- The proposed model contributes by capturing complex spatio-temporal feature characteristics of the input data [9, 10] and refining them into a non-overlapping and highly distinctive feature in embedding space [10, 21]. Additionally, the proposed model can cope with noisy data owing to orthogonal transformation of features in the model. Resulting in an improved performance of the predictive model than the existing DNN models [9, 17, 23] in this domain.
- The present model does not depend on Multi-Tasks [17] or supervised attention [19, 24] which are indispensably important in recent models for their predictive performance whereas the proposed model uses unsupervised loss for its predictive approach.
- The proposed model presents its efficacy on two real-world city taxi-demand datasets with varying experimental conditions over many popular models existing in this field.

The organization of the paper is as follows: we discuss the literature in Section 2. Section 3 outlines the problem definition in detail and is followed by preliminary keywords description in a separate section in Section 4. The methodology of our proposed approach is discussed in Section 5. Next, we see our dataset, experimental set-up, and baselines description in Section 6. Results and analysis is discussed in Section 7 followed by conclusion and future work in Section 8.

2 Related works

In this section, we review mainstream schemes on traffic flow prediction. Traffic flow prediction is of prime importance in developing smart city [25] traffic operations. However, the accurate prediction of the total traffic flow between the locations is still a significant challenge in this research area. For a decade, researchers have been improving the prediction methodology to make accurate forecasts of the vehicle's demand at a required time and place. Some well known recent predictive approaches in this field include Dynamic Tensor Completion (DTC), Region POI Demand Identification (RPDI), Ensemble method, encoder-decoder convLSTM framework and Canonical Polyadic Alternating Least Square (CP-ALS) [11, 15, 26, 27], Graph based models [9, 10] etc. to name a few.

The underlying supports of these predictive approaches can be grouped as traditional models (generally uses bayesian, Discrete fourier transform etc.), tensor based models and Deep Neural Network (DNN) models [28–30].

Next we present an elaborated view on articles for each type of predictive approach.

Traditional approaches like ARIMA, SARIMA [11, 31] and other modifications of auto-regressive approaches serve as basic models for emerging tensors and deep learning based approaches. However, such methods cannot capture the non-linearity and parallel dependency of traffic time-series. The performances of these approaches as well as statistical approaches (like Bayesian update) are quite often challenged by those using tensors [19, 32], DNN approaches [19, 33], etc. Still, these approaches set the basic guidelines for any emerging predictive model that should retain the influence of past values.

Tensor models utilises matrix factorization approach for finding the missing values in traffic ODMs [34, 35] while performing traffic demand predictions [11, 15]. Such models are data driven and produces high quality results. The matrix factorization concept has been generalized as Tensor-Decomposition. Canonical Polyadic Decomposition (CP-Decomposition), Higher-Order SVD (HOSVD), HOOI, Tucker Decomposition, etc. are few tensor decomposition approaches that have been widely used in this domain as the principle for prediction model construction. CP has used the decomposition of multi-dimensional traffic information. However, CP did not consider the network characteristics i.e. different cities might have different traffic characteristics. Dynamic Tensor Completion (DTC) [11] is another tensor model i.e a short-term traffic prediction model based on Tucker-Decomposition approach exploiting multi-dimensional information. It employs the concept of data imputation for missing values in matrix using tensor approach. However, DTC has to carry a rank determining approach being a modification of Tucker-Decomposition(QDA [36]). Additionally, DTC would always depend on an extra temporal dimension to predict traffic for minutes, it required one additional temporal dimension as 'day'. Also, similar to other tensor-based model it is not able to implement the peculiarity of different city traffic in a single model as a generic one.

Despite decent prediction performance of the Tensor-based models, DNN models provide higher flexibility to adopt auxiliary information and tap deeper latent features from time-series information. Hence DNN's are heavily explored in this field.

DNN methods Recent works mainly focus on traffic flow prediction at any particular location using neural network based techniques [4, 9, 10]. Hoang et al. [33] integrates a seasonal model, trend model, then a residual model to predict the numbers of in-flow and out-flow crowds of a region, while Zhang et al. [37] proposes a deep-learning-based approach called ST-ResNet to solve the same task. But these two studies overlook the transferring relationships among regions. Deng et al. [38] design a latent space model based on road networks to predict traffic matrix, which learns the attributes of vertices in latent spaces to capture both topological and temporal properties. LSTMs based methods are employed in many research including [2, 17, 39]. Nevertheless, they either have not considered the problem from both spatial and temporal perspectives or fail to give an adequate and meaningful representation for each region. Further graph based predictive models find its wide application in recent traffic prediction tasks. Next, we discuss significant work done in this domain.

Graph models Recent traffic prediction techniques involve graph representation learning. Hamilton et al. [40] propose GraphSAGE, a general inductive framework that leverages node feature information to efficiently generate node embeddings for graph data. Unfortunately, they just focused on the spatial perspective and cannot capture the temporal trend of the data. Further Seo et al. [41] build a model called GCRN to generalize the classical RNN to structured data by an arbitrary graph, which can be used to predict sequences of structured data. However, this method is incapable of modeling the transferring relationship between areas because they have not considered the semantic neighbors.

AttConvLSTM [19] is another DNN model which captures spatial features of only adjacent locations while spatial features of distant locations are ignored. STGCN [10] also uses GCN for spatial feature extraction and GLU (gated linear unit) for temporal feature extraction and also provides space for multi-dimensional GCN. Spatial feature extraction of graphs using GCN is the most basic form and hence there is no scope of sequential feature correlation between graphs at different time-stamps. Further STMGCN leverages the power of GCN with RNN. GCN can capture the wide range of spatial features of nearby and distant locations. However, using STMGCN traffic demand at a particular node can be predicted only and there is no provision for predicting traffic between two regions. Though it uses the most basic form of graph convolution attention is provided for better traffic prediction and it plays the role of scaling factor rather than feature space embedding.

GEML [17] gives a new direction in the traffic flow prediction task while capturing latent spatio-temporal characteristics but the model is getting trained on features using multi-task learning (MTL) other than the specific tasks, resulting in a constrained performance gain. Additionally, it has no concept of feature channeling (an organization that we implement using orthogonality constraints.) hence, overlapping features with different feature attention as well as noise leads to its poor performance. Similar to GEML, STMGCN and other trendy DNN models also lack feature organization. The spatio-temporal features [9, 10, 42] captured by the above techniques have less distinctive

capability and overlapping characteristics, which may lead to compromised predictive performance.

Generative Adversarial Network (GAN) adds another dimension to the traffic prediction task. Probabilistic Forecasting approach using Generative Adversarial Network (ForGAN) [43] is a conditional Generative Adversarial Network (GAN) model used to predict multi-step time series data. ForGAN exploits both powers of conditional GAN to generate fake time series data with the realistic distribution. However, being a generative model requires a good amount of training information which is not always available in the objectives of this domain. Also, training for ForGAN is relatively tedious and difficult.

Emerging Transformer concept has also been implemented for the said task. Graph Multi-Attention Network (GMAN) [44] aggregates both spatial and temporal traffic characteristics using multi-head attention and uses spatial and temporal embedding as attention to the series of the adjacency matrix and feature matrix. GMAN is capable to predict feature matrix for the traffic network instead of region pair (ODT) owing to its limitation of dense layer as final decoding layer for the output. GMAN being a parallel attention model is anticipated to be a noise immune model which can prevent accumulation of error over recursion when RNN, LSTM, etc. is used. However, GMAN has a limitation that for two-way traffic prediction it does not use both region's features hence to-and-fro traffic prediction is not much useful. GMAN is limited to making a prediction at a region but not across two regions or more.

Considering the pitfalls in the existing techniques, we propose a spatio-temporal DNN model ST-A$_G$P, that is capable of capturing non-lapping and highly distinctive features and hence leads to improved performance with a good margin. In the subsequent sections, we discuss the problem definition in detail followed by our proposed model description.

3 Problem statement

Typically, the traffic taxi-demand time series prediction problem requires the prediction of an ODT ($\mathcal{A} \in \mathbb{R}^{t\times n\times n}$) given different temporal snapshots of the ODMs for a city. The values in ODMs ($A \in \mathbb{R}^{n\times n}$) are the total trips from the source (n) to the destination (n) within a given time interval t. The trip information is obtained from the start and finish GPS information of the vehicles along with the timestamps. Thus a trip is represented by an ordered pair given as $[(x_s, y_s, t_s), (x_d, y_d, t_d)]$ where (x_s, y_s) and (x_d, y_d) represent the latitude, longitude pair of the source and destination respectively, and t_s, t_d represent the corresponding timestamps. The trip information at a time interval can be used to create the ODM, A_i, representing the volume of trips between different locations. Additionally, each location in an ODM (A) has some features (f) which is represented by $X \in \mathbb{R}^{n\times f}$ as a feature matrix corresponding to A. Similarly, historic ODT (H-ODT) $\mathcal{A}_H = \{A_1, A_2, \ldots, A_t\}$ having historic window size t (HWND: t), is stacked ODMs over consecutive t time intervals has its corresponding feature ODT (F-ODT) $\mathcal{X}_H = \{X_1, X_2, \ldots, X_t\}$ for the same time intervals. Given the H-ODT and F-ODT ($\mathcal{A}_H$, $\mathcal{X}_H$), the objective is to find predicted ODT (P-ODT) $\hat{\mathcal{A}}_P = \{\hat{A}_{t+1}, \hat{A}_{t+2}, \ldots, \hat{A}_r\}$ having prediction window size r (PWND: r), against the original ODT O-ODT ($\mathcal{A}_G$) that minimizes the following loss function,

$$\underset{\theta}{\text{argmin}} \sum_{i=t+1}^{r} \|A_i - \hat{A}_i\|_2,$$

where $\hat{A}$, A is one of the predicted ODM and corresponding original ODM respectively. $\boldsymbol{\theta}$ is the model parameter. We next discuss the proposed model in detail.

4 Preliminary

In this section, we discuss preliminary concepts and important terminology that would help further understanding the proposed model and intuition behind the proposed approach.

Principal Component Analysis (PCA) & Condition Number: PCA is a dimensionality reduction approach which is a transformation of information into a non-overlapping (orthogonal) and highly distinctive spectral space called principal components. It facilitates introducing orthogonality by principal components and high variance of the information along with these components (Fig. 1a). Provided, $\mathcal{F}_{n\times f}$ being f features matrix of n instances and $\Theta_{f\times d}$ is a matrix of d principal components. Then columns of Θ are principal components being orthogonal to each other ($\Theta'\Theta = \mathcal{I}$) and features in $\mathcal{F}$ would have high variance along with these components. The transformed feature matrix $\mathcal{R}_{n\times d}$ is obtained by:

$$\mathcal{R} = \mathcal{F} * \Theta \tag{1}$$

Where information in $\mathcal{R}$ are now non-overlapping and distinctive. Additionally, on applying matrix multiplication with Θ, the noise in $\mathcal{F}$ is not magnified as *condition number of an orthogonal matrix is low* [45, 46], a less known facts about orthogonal matrix transformation.

$$\mathcal{R} + \delta r = (\mathcal{F} + \delta F) * \Theta \tag{2}$$

Where ratio of δr and δF would be 1 if the matrix Θ is orthogonal. That means, the noise (perturbation) is not magnified in this case but any other matrix transformation.

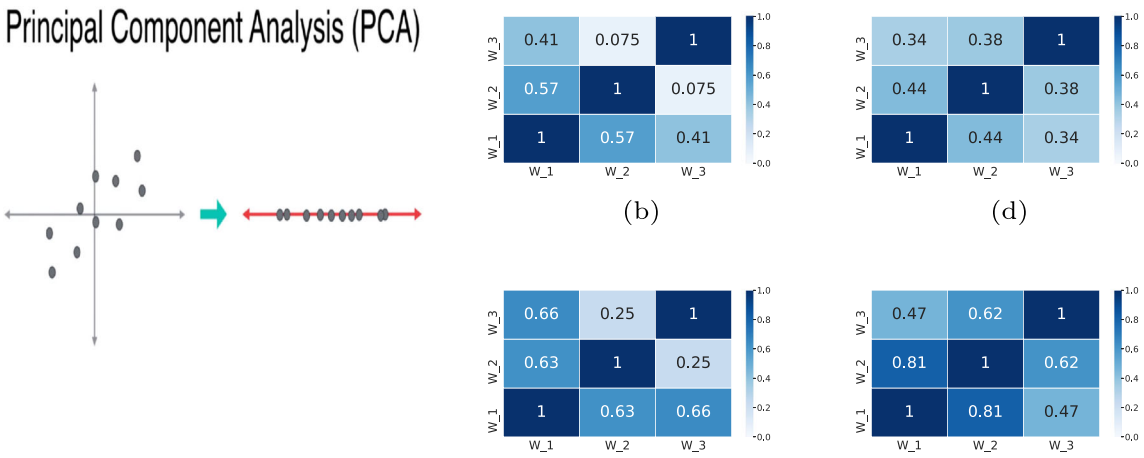


Fig. 1 PCA (Fig. 1a): Illustrating that the variance of projected data on Principal component (red line) increases as compared to x, y axes. Heatmaps (Figs. 1b, 1d 1c, 1e) on inner product of kernel with (PCA$_G$, up) and without (PCA$_G$, down) unsupervised loss. Observing, the values off the Counter-diagonals in with PCA$_G$ are relatively lower than that of without PCA$_G$, indicating that the kernels approaching 0 (orthogonal) with unsupervised loss

The present approach attempts to analogize these properties by transforming historic ODT information (spatio-temporal characteristics) onto orthogonal embedding space with high variance, which creates highly distinctive embedded features. Moreover, during prediction, the transformation of spatio-temporal characteristics through the orthogonal components (PCA$_G$, Section 5.3.2) of the model prevents high propagation of noise from input information to the yield. Hence, the prediction results are immune to noise and it's worth desired property of any predictive model. Consequently, the proposed approach creates a distinguishable spatio-temporal feature for better prediction of taxi-demand time series across different locations in a city.

Graph Convolution Network (GCN): Graph Convolution Network is a better choice over Convolutional Neural Network when the input sample is a graph and the connectivity between nodes has an important role however far the nodes are. The graph convolution network helps embed node characteristics based on the node's features as well as the aggregated features of its neighboring nodes. Hence, unlike the grid structure, two locations even geographically far can be well captured by a GCN if there is connectivity between the locations while a CNN would most likely miss this relationship. *GCN provides similar embedding for the nodes with similar neighbors characteristics. Hence, GCN brings contextually similar nodes close in embedding space*. The two forms of graph convolution are (i)*Spatial Graph Convolution* and (ii)*Spectral Graph Convolution*, which we describe below using a graph having n nodes and each node having f features where $A_{n\times n}$ and $X_{n\times f}$ represent adjacency matrix and feature matrix of the corresponding graph:
Spatial Graph Convolution: Spatial graph convolution creating a node embedding by aggregating the node's features and its neighboring nodes' features as follows:

$$H = \sigma(AXW + b) \tag{3}$$

Where $H \in \mathbb{R}^{n\times d}$ being graph embedding after convolution. σ is an activation function. $W \in \mathbb{R}^{f\times d}$ is GCN kernel and $b \in \mathbb{R}^d$ is bias term. The overall procedure is a transformation of graph A with features X from f dimension to a d dimensional space.

Spectral Graph Convolution: Spectral convolution helps to transform a graph embedding from node domain to spectral domain. Spectral domain being orthogonal helps to override the redundancy in the graph embedding. Replacing the computationally costly spectral decomposition, spectral graph convolution is carried using the coefficients of chebyshev polynomial of first degree. Below, x is a signal (features in node domain) is transformed to spectral domain as follows:

$$g_\theta \star x = U g_\theta U^T x \tag{4}$$

Where U is spectral decomposition of normalized graph Laplacian $L = \mathcal{I}_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ and $g_\theta = diag(\theta) \in \mathbb{R}^n$ is parameterised by θ. Spectral decomposition being expensive, is replaced by coefficients of Chebyshev Polynomial of first order as:

$$g_\theta \star x \approx \sum_{k=0}^{K} \theta_k T(\hat{L}) x \tag{5}$$

Where $\hat{L} = \frac{2}{\lambda_{max}} L - \mathcal{L}_n$ with λ_{max} being highest eigen value and θ_k is Chebyshev coefficients of order K. Under

approximation for first order Chebyshev polynomial, the above equation can be reduced to:

$$g_\theta \star x \approx \theta_k(\mathcal{I}_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x \tag{6}$$

Hence a spectral graph convolution can be approximated by a GCN as follows:

$$H = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X\theta \tag{7}$$

Where $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ is an approximation of normalized reduced Laplacian of A and X is a feature matrix while θ is GCN kernel which is learned during training. Hence (7) is spectral graph convolution and been proved to be very useful for convolution in many literatures [18, 47].

5 Proposed methodology

In this section, we provide a detailed description of proposed methodology. A schematic view of the operational steps taken by the proposed model can be seen in Algorithm 1 and Figs. 3 and 4. We start with describing the characteristics of taxi-demand time-series that includes spatio-temporal characteristics.

5.1 Characteristics of Taxi-Demand Time Series in a City

Taxi-Demand of various regions of a city over a period of time forms a group of correlated time-series data with some specific characteristics [29, 31, 48]. In this section, we would define two of these important characteristics which can be very helpful for the prediction. We term this as spatio-temporal characteristics. Additionally Table 1 list out all the important symbols used in our proposed model.

Spatio-Temporal Characteristics: Most often, there are many different kinds of spatio-temporal characteristics of demand traffic time series. The two prominent characteristics [17] are (i) *proximal characteristics* and (ii) *mobility characteristics* which are greatly involved in regulating the demand between two locations. *Proximal Characteristics* [17] relates to the demands based on the distance of the regions like often two nearby city regions would experience more traffic transition than outskirt regions of the city. Similarly, *Mobility Characteristics* [17] relates to those regions which are purposely connected as residential and office regions. Next, we describe each of these formally:

- **Proximal Characteristics:** Depending upon the distance between locations, demand among them most often follows an inverse trend. Nearby locations are found to have more traffic demand transitions among them than that among the distant locations. We define this proximal characteristics as $\mathcal{C}_p \in \mathbb{R}^{t \times n \times n}$ where a cell value in $C_{n \times n} \in \mathcal{C}_p$ is defined as follows:

$$c_{x,y} = 1 - \frac{dist(x, y)}{\sum_{z \in Nei(x)} dist(x, z)} \tag{8}$$

  where x, y are two locations and $Nei(x)$ defines all neighbors of x. $dist(x, y)$ returns euclidean distance between the centres of the locations x and y.
- **Mobility Characteristics:** Sometimes few locations are having some specific characteristics, and despite being very distant, these locations are having a continuous traffic (taxi-demands) transition with the others. For an example, airport in a city is located at a distant place, still it has high traffic exchange with few commercial and residential regions. Consequently, demand traffic between two locations is also affected by the volume of demand traffic between them. We define these characteristics as "mobility" between the locations. It is presented by $\mathcal{C}_m \in \mathbb{R}^{t \times n \times n}$ where $C_{n \times n} \in \mathcal{C}_m$ has a value in cell as follows:

$$c_{x,y} = \frac{mob(x, y)}{\sum_{z \in Nei(x)} mob(x, z)} \tag{9}$$

  Similarly, $x, y, Nei(x)$ are defined above. $mob(x, y)$ returns the demand traffic volume between the locations x, y in a given time interval t when $c \in C_{n \times n}$ at time interval t.

Table 1 Some of important symbols used in explaining the proposed model

Symbols	
$\mathcal{A}, \mathcal{X}, \mathcal{C}$	Tensors (Origin-Destination, Feature, Characteristics)
A, X, C	Matrices (Origin-Destination, Feature, Characteristic)
$[\hat{A}_{t+1}, \hat{A}_{t+2} \ldots \hat{A}_r]$	A sequence of Predicted ODMs
$H, \widetilde{H}, \widetilde{H}_{agg}$	Node Feature Representation, Embedded ODM Representation (GCN, PCA_G)
θ_s, θ_d	parameters: spatial and predictor
p, m, cb	Subscripts: proximal, mobility, spectral convolution
t, r	Historic Window size, prediction window size
$\oplus, \alpha$	Aggregation operation: concatenation, addition, characteristics attention
$\mathcal{L}_o, \mathcal{L}_v$	Unsupervised Loss: orthogonal, variance

These two spatio-temporal characteristics ($\mathcal{C}_p, \mathcal{C}_m$) are used as attention in the proposed model. *Importantly, these mentioned characteristics do not require any additional data but can be easily derived from the historic ODT in use*. We next elaborate on the procedural steps for the construction of the proposed model.

5.2 Proposed model

Based on primary operations, the proposed model is having two operational modules (i)*Aggregator*, (ii)*Predictor* which acts in coordination to achieve refined prediction results for P-ODT. A conceptual view of component's operation is visualized in Figs. 3 and 4. We start by briefly describing these two modules and then followed by a detailed description of the proposed model.

- **Aggregator:** This operational module is involved in aggregating input time series features in H-ODT into an enhanced spatio-temporal embedded features using components like: *Graph Convolution Network* (**GCN**$_F$ or **GCN**$_{cb}$), *Principle Component Aggregator* (**PCA**$_G$) and *Long Short-Term Memory* Recurrent Neural Network (**LSTM**). These components help in creating a refined feature representation for the input from varying sources having different feature characteristics.
- **Predictor:** This operational module is used to predict the aggregated embedded representation of H-ODT into the final multi-step predicted Origin-Destination Tensor P-ODT. The module consists of a *Predictor* (**P**$_r$**D**) component. A single Predictor kernel is used to obtain multiple P-ODMs to learn dominant temporal behavior across ODMs.

The above components are used in our model construction in a combination. The proposed model is a combination of three models (i) Proximal model: **GCN**$_F$**+LSTM+P**$_r$**D** (ii) Mobility model: **GCN**$_F$**+LSTM+P**$_r$**D** (iii) ChebNet model: **GCN**$_{cb}$**+PCA**$_G$**+LSTM+P**$_r$**D**. Consequently the proposed model has 3 input terminals as well as 3 output terminals. The **PCA**$_G$ component of ChebNet model receives inputs from Graph convolution components of all the three models. **LSTM** component in each model receives inputs from its predecessor component in their respective model explicitly. We illustrate our model in Figs. 3 and 4 and its simplified procedural steps in Algorithm 1. Next, we elaborate each components of the proposed model and provide the necessary procedural[1] steps taken into each of them.

[1] A common operation over each element of a sequence is presented using an arrow with common operands and operation denoted above and below arrow sign.

5.3 Aggregator

This module transforms the input samples from node-domain feature space to embedding-feature space using *Graph Convolutional Network* GCN$_F$ (GCN$_{cb}$). GCN$_{cb}$ is spectral graph convolution with no explicit characteristics. Provided historic information based on different characteristics (C_i) as denoted by $\mathcal{A} \in \mathbb{R}^{t\times n\times n}$, $\mathcal{X} \in \mathbb{R}^{t\times n\times f}$ and $\mathcal{C} \in \mathbb{R}^{t\times n\times n}$ where $\mathcal{C}$ denotes a specific characteristics of historic information, we elaborate working of Graph Convolution (GCN$_F$) next.

5.3.1 Featured graph convolution network (GCN$_F$):

GCN$_F$ works on a sequence of $\{A_j\} \in \mathcal{A}_i, \{X_j\} \in \mathcal{X}_i$ and $\{C_j\} \in \mathcal{C}_i$ as follows:

$$[\{A_1, X_1, C_1\}, \{A_2, X_2, C_2\}, \ldots \{A_t, X_t, C_t\}] \xrightarrow[\theta_s,k]{GCN_F} [\widetilde{H}_1, \widetilde{H}_2, \ldots \widetilde{H}_t] \quad (10)$$

The GCN$_F(\theta_s, k)$ operates on each triplet of $\{A_i, X_i, C_i\}$ at i^{th} time interval as follows:

$$\{A_i, X_i, C_i\} \xrightarrow{k} \{[A_i^1, A_i^2, \ldots A_i^k], X_i, C_i\} \quad (11)$$

$$\{[A_i^1, A_i^2, \ldots A_i^k], C_i\} \xrightarrow{X_i} \{[H_i^1, H_i^2, \ldots H_i^k], C_i\} \quad (12)$$

$$\{[H_i^1, H_i^2, \ldots H_i^k]\} \xrightarrow{\alpha(C_i)} \{[H_{\alpha i}^1, H_{\alpha i}^2, \ldots H_{\alpha i}^k]\} \quad (13)$$

$$\{[H_{\alpha i}^1, H_{\alpha i}^2, \ldots H_{\alpha i}^k]\} \xrightarrow{\theta_s} \{[\widetilde{H}_i^1, \widetilde{H}_i^2, \ldots \widetilde{H}_i^k]\} \quad (14)$$

$$\{[\widetilde{H}_i^1, \widetilde{H}_i^2, \ldots \widetilde{H}_i^k]\} \xrightarrow{\sum} \{\widetilde{H}_i\} \quad (15)$$

Where $\theta_s \in \mathbb{R}^{f\times d}$, $H \in \mathbb{R}^{n\times f}$ and $\widetilde{H} \in \mathbb{R}^{n\times d}$ are having f node features and d embedding dimension. α is characteristics attention. The attention is provided based on the present value of the node and its neighbors. For example, if n_i is a node having a particular value $h_i \in H$ and $c_i \in C$ and correspondingly the same is true for each of its neighbors n_j as $h_j \in H$ and $c_j \in C$, then the characteristics attention provided by GCN$_F$ to the node n_i is as follows:

$$h_i = c_i h_i + \sum_{j}^{Nei} \frac{\|c_j\|}{\|\sum_{l}^{Nei} c_l\|} h_j \quad (16)$$

The above equation in matrix notation can be expressed as $H \xrightarrow{\alpha(C)} CH$ which is spatial aggregation of a node value based on its weighted neighbors values. Hence for each spatio-temporal characteristics GCN$_F$ produces spatially aggregated representation as follows:

$$\{A_i, X_i, C_i\} \xrightarrow[\theta_s,k]{GCN_F} \{\widetilde{H}_i\}_c \quad (17)$$

Here subscript c represents different spatio-temporal characteristics of the input tensor which can be replaced with p, m referring to spectral graph convolution with *proximal*, *mobility* characteristics and *cb* for simply

Algorithm 1 ST-A$_G$P Framework

Input: $\mathcal{A} \in \mathbb{R}^{t\times n\times n}, \mathcal{X} \in \mathbb{R}^{t\times n\times f}, \mathcal{C}_p \in \mathbb{R}^{t\times n\times n}, \mathcal{C}_m \in \mathbb{R}^{t\times n\times n}, s, v$ // Input ODT, Feature ODT, Proximal characteristics ODT, Mobility characteristics ODT, Test split ratio and validation split ration respectively.

Output: $\hat{\mathcal{A}}_s \in \mathbb{R}^{r\times n\times n}$ // Prediction ODT

Function ST-A$_G$P$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m)$:

$\{\widetilde{H}_p\} \leftarrow GCN_F(\mathcal{A}, \mathcal{X}, \mathcal{C}_p)$ // Spatial Aggregation with proximal characteristics

$\{\widetilde{H}_m\} \leftarrow GCN_F(\mathcal{A}, \mathcal{X}, \mathcal{C}_m)$ // Spatial Aggregation with mobility characteristics

$\{\widetilde{H}_{cb}\} \leftarrow GCN(\mathcal{A}, \mathcal{X})$ // Spatial Aggregation using chebNet

$\{\widetilde{H}_{agg}\} \leftarrow PCA_G\left(\left[\{\widetilde{H}_p\}, \{\widetilde{H}_m\}, \{\widetilde{H}_{cb}\}\right]\right)$ // Aggregation Operations as concatenation or addition

$\hat{\mathcal{A}}_p \leftarrow P_rD(LSTM(\{\widetilde{H}_p\}))$ // Temporal Aggregation and Prediction

$\hat{\mathcal{A}}_m \leftarrow P_rD(LSTM(\{\widetilde{H}_m\}))$ // Temporal Aggregation and Prediction

$\hat{\mathcal{A}} \leftarrow P_rD(LSTM(\{\widetilde{H}_{agg}\}))$ // Temporal Aggregation and Prediction

return $\hat{\mathcal{A}}, \hat{\mathcal{A}}_p, \hat{\mathcal{A}}_m, \{\widetilde{H}_p\}, \{\widetilde{H}_m\}, \{\widetilde{H}_{cb}\}$

Function PREDICT$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m)$:

$\hat{\mathcal{A}}, \hat{\mathcal{A}}_p, \hat{\mathcal{A}}_m, \{\widetilde{H}_p\}, \{\widetilde{H}_m\}, \{\widetilde{H}_{cb}\} \leftarrow$ ST-A$_G$P$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m)$

return $\hat{\mathcal{A}}$ // Returning the predicted ODT of the main model.

Function UPDATE$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m, v)$:

return Loss(ST-A$_G$P$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m), v)$ // Update on training information and return evaluation loss on validation information

Function TRAIN-n-PREDICT$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m, s, v)$:

$\mathcal{A}_s, \mathcal{X}_s, \mathcal{C}_p^s, \mathcal{C}_m^s \xleftarrow{s} \mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m$ // extract test data and reduce original data by $(1-s)$ split ratio.

while !*loss converges* **do**

loss = UPDATE$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m, v)$

end

return PREDICT$(\mathcal{A}_s, \mathcal{X}_s, \mathcal{C}_p^s, \mathcal{C}_m^s)$

TRAIN-n-PREDICT$(\mathcal{A}, \mathcal{X}, \mathcal{C}_p, \mathcal{C}_m, s, v)$

graph spectral convolution with no characteristics input (i.e. skipping (13)). We denote the parameters of graph convolution θ_s (GCN$_F$ or GCN$_{cb}$) with W (kernel weight) with a numeric subscripts as represented in Figs. 1b and d in order of *p, m* and *cb*.

5.3.2 Principle component aggregator (PCA$_G$)

GCN$_F$ brings about the orthogonal projection (spectral convolution, Section 4) of the node features along with providing attention to the nodes based on their spatio-temporal characteristics. PCA$_G$ operates similarly by providing orthogonal projection of varying spatio-temporal characterized GCN outputs and aggregates them together to have non-overlapping embedding. Additionally, PCA$_G$ attempts to introduce distinguishable embedding representation between nodes by using variance maximization. PCA$_G$ works on the sequential outputs of the 3 Graph Convolution Networks as follows:

$$\omega_1[\widetilde{H}_1, \ldots \widetilde{H}_t]_p \oplus \omega_2[\widetilde{H}_1, \ldots \widetilde{H}_t]_m \oplus \omega_3[\widetilde{H}_1, \ldots \widetilde{H}_t]_{cb} \xrightarrow[\mathcal{L}_o, \mathcal{L}_v]{PCA_G} [\widetilde{H}_1, \ldots \widetilde{H}_t]_{agg} \tag{18}$$

Where $\oplus$ is an aggregation operation (concatenation, addition) and ω_i is i^{th} co-efficient implemented using dense layer. It operates on each temporal slice and aggregates their feature together as follows:

$$\omega_1\widetilde{H}_p \oplus \omega_2\widetilde{H}_m \oplus \omega_3\widetilde{H}_{cb} \xrightarrow[\mathcal{L}_o, \mathcal{L}_v]{PCA_G} \{\widetilde{H}\}_{agg} \tag{19}$$

Where $\widetilde{H}_i$ belongs to $\mathbb{R}^{n\times d}$ but $\widetilde{H}_{agg}$ might have either of these two space $\mathbb{R}^{n\times 3d}$ (concatenation) or $\mathbb{R}^{n\times d}$ (addition) depending on aggregation operation used. The unsupervised losses ($\mathcal{L}_o$, $\mathcal{L}_v$) are computed at this point. The term unsupervised is used since such loss does not require explicit labeling. $\mathcal{L}_o$ and $\mathcal{L}_v$ is explained later, and they depend on the following inputs :

$$minimize \quad \mathcal{L}_o = f(\widetilde{H}_p, \widetilde{H}_m, \widetilde{H}_{cb}) \tag{20}$$

$$maximize \quad \mathcal{L}_v = f(\widetilde{H}_i) \quad i = \{p, m, cb\} \tag{21}$$

5.3.3 LSTM:

It is a variation of RNN that can capture long-term dependency in sequence and is able to cope up with gradient problems. LSTM [17, 49] has been very much efficient in many sequence predictions over RNN and it operates using a gated mechanism to produce its output. The following shows the operational steps of different gates of an LSTM cell where x_t ($\widetilde{H}_{agg^t}$) is the input to LSTM cell at t^{th} timestamp:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{22}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{23}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{24}$$

Where i, f, o corresponds to input-gate, forget-gate and output-gate in the t^{th} LSTM cell. w and b are the weight of the corresponding gates nodes and bias respectively. h_{t-1} is the output of $(t-1)^{th}$ LSTM cell. The following equations describe the generation of LSTM intermediate cell state ($\widetilde{c}$), cell state (c) and final output (h):

$$\widetilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{25}$$

$$c = t_t * \widetilde{c}_{t-1} + f_t * c_{t-1} \tag{26}$$

$$h_t = o_t * \tanh(c_t) \tag{27}$$

In the above equations, (26) forms a deciding factor where LSTM keeps only the relevant information of the previous state and forgets the rest. The sequence of the aggregated output from the previous components is received by LSTM to capture temporal dependencies between encoded features of the nodes, where LSTM captures those dependencies and produces a temporally aggregated sequence:

$$[\widetilde{H}_1, \widetilde{H}_2, \widetilde{H}_3 \dots \widetilde{H}_t] \xrightarrow[r]{LSTM} [\widetilde{H}_{t+1}, \widetilde{H}_{t+2} \dots \widetilde{H}_{t+r}] \tag{28}$$

The symbols have usual meanings and previously defined dimensions. To delve more into this transformation and elaborate how the input and output of the LSTM is encoded and decoded, the following equation shows the required procedure:

$$[\widetilde{H}_1, \widetilde{H}_2, \widetilde{H}_3 \dots \widetilde{H}_i] \xrightarrow{LSTM} \widetilde{H}_{t+r-(t-i)} \tag{29}$$

The above equation creates multiple instances for each $i \in \{1, 2, \dots t\}$, of which the accepted outputs are $i \in \{t-(r-1), t-(r-2) \dots t\}$. Also, as $\widetilde{H}$ is a two dimensional tensor, it is reshaped to 1-D tensor before and after the transformation. Provided a multi-step prediction scalar r, only the final r outputs are passed to the next Predictor component where finally interpretable values of nodes are obtained as predicted traffic counts. The LSTMs in Proximal and Mobility models receive inputs from their respective GCN$_F$ components whereas LSTM in ChebNet Model receives inputs from PCA$_G$. LSTM output is passed to the predictor component in each model.

5.4 Predictor

The input to Predictor (P$_r$D) is the aggregated spatio-temporal embedding representation of features of H-ODT. Predictor uses a single kernel ($\theta_d \in \mathbb{R}^{d\times d}$) on each of the elements ($\widetilde{H}_i \in \mathbb{R}^{n\times d}$) of temporal input and predict the corresponding ODMs ($\hat{H}_i \in \mathbb{R}^{n\times n}$). The Predictor works as follows:

$$[\widetilde{H}_1, \widetilde{H}_2 \dots \widetilde{H}_r] \xrightarrow[\theta_d]{P_r D} [\hat{A}_1, \hat{A}_2 \dots \hat{A}_r] \tag{30}$$

The θ_d slides on each input and operates as follows on each individual input element:

$$\hat{A} = \widetilde{H}_i \theta_d \widetilde{H}_i^T \tag{31}$$

In the above equation T stands for matrix transpose operation. Thus a sequence of multi-step (r) predicted ODMs ($\hat{A}_i$) is obtained which is further used to estimate the error and optimize the model using the ground truth ODMs. We next define the loss used for the optimization of the model.

5.5 Optimization strategy

To optimize the model parameters, the formulated model loss is defined in this subsection. Being predictive model, the model calculates mean squared error (mse) during optimization over the prediction error. The following presents ($\mathcal{L}_{mse}$) of a single predicted ODT.

$$\mathcal{L}_{mse} = \frac{1}{n}\sum_{i}^{n}(Y_i - \hat{Y})_i^2 \tag{32}$$

The two losses defined as unsupervised loss are orthogonal loss ($\mathcal{L}_o$) and var-loss ($\mathcal{L}_v$). The orthogonal loss $\mathcal{L}_o$ is defined for each pair of embedded feature representation (e.g. $\widetilde{H}_j$, $\widetilde{H}_k$ where $j \neq k$) at the i^{th} temporal instance. The orthogonal loss is defined as follows:

$$\mathcal{L}_o = \sum_{j,k}\langle \widetilde{H}_j, \widetilde{H}_k \rangle_F \tag{33}$$

$\langle , \rangle_F$ denotes Frobenius inner-product hence making (or attempting to make) $\widetilde{H}_j$, $\widetilde{H}_k$ orthogonal (non-overlapping as much as possible) to each other, minimizing redundancy between them. Here subscripts j, k denote p, m, cb. Another loss $\mathcal{L}_v$ is introduced to maintain variation among the nodes $n \in \widetilde{H}$ and $\widetilde{H} \in \mathbb{R}^{n \times d}$ and defined as follows:

$$var = \frac{1}{N-1} \sum_{i}^{n} (n_i - n_{mean})^2 \tag{34}$$

$$\mathcal{L}_v * \sigma(var) = 1 \tag{35}$$

Where minimizing $\mathcal{L}_v$ maximizes variance (var). n_{mean} is mean of all nodes embedding in a $\widetilde{H}_{p/m/cb}$. σ is the sigmoid activation and $*$ here is scalar product. The complete model loss including 3 models is defined as follows:

$$\mathcal{L}_{final} = \Omega_1 \mathcal{L}^p_{mse} + \Omega_2 \mathcal{L}^m_{mse} + \Omega_3 \mathcal{L}^{cb}_{mse} + \Omega_4 \mathcal{L}_o + \Omega_5 \mathcal{L}_v \tag{36}$$

Where superscripts and subscripts have usual meaning and Ω_1, Ω_2, Ω_3, Ω_4 and Ω_5 are weights of each loss determined while training. Heatmaps in Figs. 1b and d reveal that unsupervised loss results into shifting of kernel weights (W) towards being orthogonal. Values off the Counter-diagonals in heatmaps are tending to 0 (not exactly 0 i.e. approaching to be orthogonal) as $\mathcal{L}_{final}$ does not only comprise of unsupervised loss but losses for other tasks too. Hence, for the best prediction results, maximum variance and non-overlapping of embedded features are obtained at these values of W (Heatmaps 1b and 1d), supporting the prime objective of ODT predictions during training. The coming section discusses the details of experimental settings.

6 Experiment

In this section, we outline the details of the experiments including the dataset and the baselines. Having taxi dataset, we form ODT for each source-destination pair, apply our proposed prediction technique to finally predict the taxi demand for each pair. We also mention the varying conditions for each experiment with the hyper-parameter settings involved. We start with data preparation for different experimental ODT creations.

6.1 Dataset

Many of the existing works have evaluated their model performance using the traffic data of two major cities [19, 23, 50, 51], the Green-Taxi of New York City [2](NYC) (Jan – Mar 2014) and private taxi-data of Thessaloniki City [3](THS) (Jan – Mar 2015). Thessaloniki city data has the records of only 3 months traffic data, so to maintain consistency in the time and volume of data, we used only 3 months traffic records of NYC, even though the data for one year is available. A trip in both NYC and THS contains the GPS coordinates (latitude & longitude) of the start and end locations of a journey. Consequently, the NYC and THS datasets have trip information of about 2.1 and 1.7 million made for three consecutive months.

6.2 Model input preparation

Now we write details of ODT formation from trip counts information with the features involved. We also elaborate on experimental setups and hyper-parameters values during experiments.

ODT Construction: Provided the GPS data of the city for 3 months time duration, we created an hourly temporal snapshot of the city traffic. Initially, the city is bounded in a rectangular box by four extreme GPS points in the dataset. Then the city as the rectangular box is divided into equal-sized grids. The center GPS value of a grid is termed as *Grid-Representative* (GR). Traffic is recorded if it starts from a grid (source) and ends in another (destination) over a given time. Grid clustering with extreme points would result in many vacant and insignificant grids [17, 23, 52]. Hence, we select only the actively participating n grids which account for top major mobility counts. For each hour, we prepare a matrix ($n \times n$) where each cell containing the count of taxi-demand traffics from a source (row) to a destination (column) in that hour [17, 19], which is called the Origin-Destination Matrix ODM ($A_i \in \mathbb{R}^{n \times n}$). These ODMs contain both intra and inter regions demands. Stacking such temporal ODMs over chosen time duration (t) forms Origin-Destination Tensor ODT ($\mathcal{A}_i \in \mathbb{R}^{t \times n \times n}$). In our present experiments, the obtained active grids (nodes) for NYC and THS are 55 and 25 respectively with a grid size of 5×5 $sqkm$ in each. The temporal slice of 2160 ODMs over 3 months from January to March are stacked to obtain respective ODT.

ODM Features/Characteristics: ODM features are defined based on the grid connectivity and mobility (taxi-demand) among the grids for providing attention to the proposed model. As discussed in Section 5, depending upon the normalized distance of the neighboring active grids representatives, proximal characteristics are calculated. Mobility characteristics are measured using the taxi-demand count between the grids over time. These features correspond to the connective links between grids. Similarly, for a given time-interval each grid (g_i as a node) has two features (f) of traffic: (a) Demand - the count of total traffic originating in any other neighboring grids which have grid g_i as destination; (b) Supply - the count of total traffic starting from the grid g_i going to any of its connected grids.

[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[3] Kaggle "Taxi Fare Challenge", 2017

6.3 Training steps, parameters & deployment

We use 80% of ODT for training purposes, out of which 20% is used for validation. The rest 20% is used for testing. Using historic ODMs of $HWND = \{12, 24, 168\}hr$, we predict next r ODMs of one hour interval, where $r \in PWND = \{1, 2, 4, 6, 8, 10, 12\}hr$. In our experiments, we further use a min-max scaling, grid-search for tuning, and relu, sigmoid, tanh as activation functions, 0.2 dropout rate, 500, 793 epochs for NYC, THS respectively, and $L1(10^{-6})$ regularizer. Dimension size of reported results is 10 with depth $k = 2$ in GCN$_F$ and each Ω has value 1.0. The aggregation operation is *concatenation*. Mean Squared Error is used as training loss with $adam(10^{-4})$ optimizer and *accuracy* is training metric. The reported results are on the predicted outputs of ChebNet Model. The model architecture can be seen in Fig. 2 illustrating the flow of input to output. The experiments are done on a system with the following details: Intel i5, 3.5GHz, 16GB RAM. The programming language used is Python. Some of the most used libraries are Scipy, Folium, Igraph, and Keras, Tensorflow, Pandas, Numpy, Matplotlib, etc. The experiment is carried on a GPU system with 11GB NVIDIA GeForce GTX-1080 Ti-GPU configuration. Next, we detail the experimental setup for each of the major steps outlined in our approach (Figs. 3 and 4).

6.4 Competitive systems

For the performance comparison, we have considered trendy predictive models in this domain. Majority of which are the DNN models having similar experimental conditions, while we have used a DNN generative model as well as a Tensor based model for a wide range of testing platforms of the proposed model. Ultimately, we have compared the proposed model with a few of its variants to identify the effectiveness of each of the components in the proposed model.

Deep neural network models : GEML [17] and **STMGCN** [9] are two trendy models in this category that have shown their high performance deliverance in this application field. GEML uses an MTL approach with two traffic characteristics for attention and uses GCN and LSTM for prediction. STMGCN uses temporal attention with multi-graph convolution using GCN and RNN for prediction. STMGCN uses chebNet in its model.

Generative adversarial networks : ForGAN [43] is a conditional GAN model used for time series prediction. It uses historic information as a condition and is efficient in learning the data distribution of the input time series.

Tensor decomposition based model : TeDCaN [16] is a tensor based model which incorporates network characteristics for multi-step prediction. As the experimental setup of TeDCaN varies widely with that of mentioned DNN models, we compare TeDCaN ($weeks = 2, k = 70$ [16]) with the proposed model for multi-step prediction at historic window of $hr = 24$ only. Reports are presented separately.

Variations of the proposed models : The proposed model is compared to its different variations in order to study the effect of each component of the model.

ST-A$_G$P~P: This variant is created by removing the Proximal model from the main model. It can evaluate the contribution of Proximal characteristics in multi-step prediction.

ST-A$_G$P~M: This variant helps us to understand the importance of Mobility characteristics in the proposed model. It is constructed by omitting the Mobility model from the proposed model structure.

ST-A$_G$P~PCA$_G$: This variant is created by omitting PCA$_G$ component from ST-A$_G$P. This can help us understand the impact of feature channeling (orthogonality) and noise immunity (unsupervised attention) in the proposed model.

ST-A$_G$P~ $\mathcal{L}_{oss}$: By omitting $\mathcal{L}^p_{mse}$ and $\mathcal{L}^m_{mse}$ from the proposed model, this variant is created. This helps us study the effect of optimizing each component model in the main model.

ST-A$_G$P~Cheby: This variant is prepared by replacing spectral graph convolution with spatial graph convolution in the model. This offers us to understand whether spectral

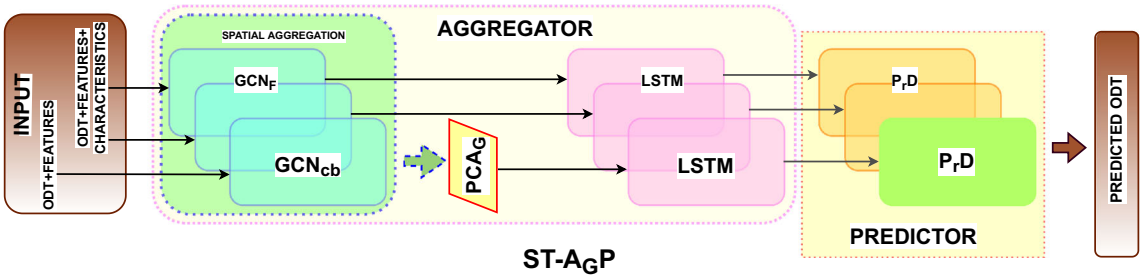


Fig. 2 Block Diagram of the proposed architecture ST-A$_G$P. The inputs are provided to 'Aggregator' followed by 'Predictor'. Brown blocks are input-output on left-right ends respectively

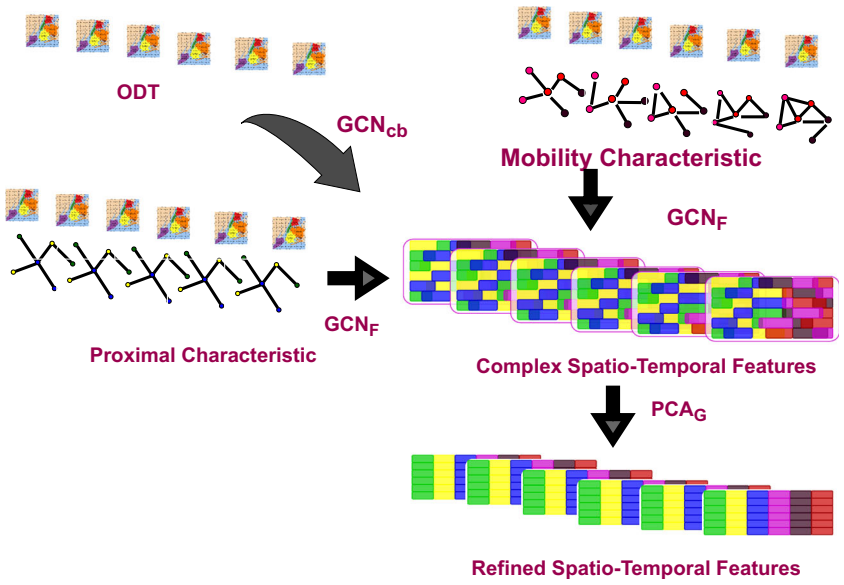


Fig. 3 Historic ODT and Characteristics Networks are taken as input to GCN and GCN$_F$. The aggregated feature is passed to PCA$_G$. PCA$_G$ operates on unorganized features to eliminate overlapping, feature creeping, reduce noise so that the best of the features can be used for prediction task

decomposition has a noticeable contribution in multi-step prediction of taxi-demand time series.

6.5 Evaluation measures

Following standard literature in this domain [9, 17], evaluation of the proposed model is carried for the multi-step prediction of the hourly ODMs on the test data using root mean square error (*RMSE*), mean absolute percentage error (*MAPE*), and symmetric mean absolute percentage error (*SMAPE*). These values are derived using the multi-step predictions ($\hat{\mathcal{A}}_P \in \mathbb{R}^{r\times n\times n}$) and the true values ($\mathcal{A}_G$) regarding the chebNet Model containing PCA_G. The proposed model is evaluated for the model performance under (i) varying historic window size (ii) under the varying percentage of missing information in the historic data (iii) for varying dimension size of the model and aggregation operations in PCA$_G$. The formulae are represented by element (cell value) $\hat{y} \in \hat{\mathcal{A}}_P$ and $y \in \mathcal{A}_G$ as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{r\times n\times n}(y_i - \hat{y}_i)^2}{r \times n \times n}} \tag{37}$$

The other measure that we use for performance evaluation is the mean absolute percentage error (*MAPE*). MAPE is a statistical measure used to determine the accuracy of a forecast and is given as

$$MAPE = \frac{1}{r \times n \times n}\sum_{i}^{r\times n\times n}\frac{|y_i - \hat{y}_i|}{|y_i|} \tag{38}$$

Unlike the RMSE, MAPE [9, 53] represents the error in terms of the percentage with respect to the true value,

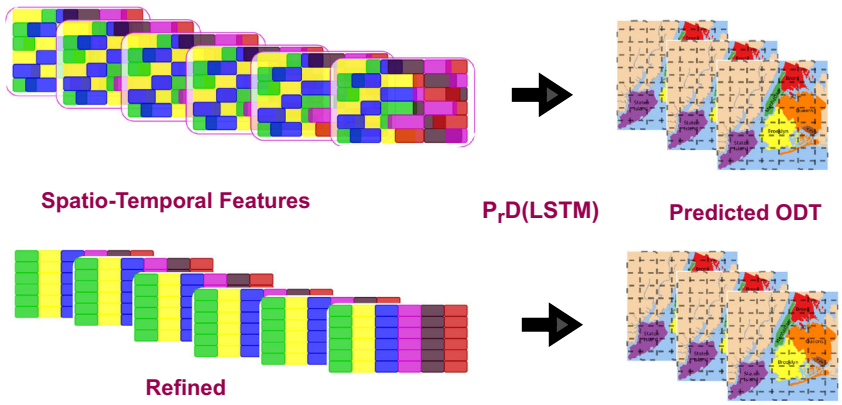


Fig. 4 Feature representation used for prediction without PCA$_G$ (above) and with PCA$_G$ (below) are shown. Evidently, this is why PCA$_G$ module reveals better prediction performance

SMAPE presents the symmetric measure of the results for boundedness and is also helpful metric for different datasets [17].

$$SMAPE = \frac{1}{r \times n \times n} \sum_{i}^{r \times n \times n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)} \tag{39}$$

The reported values of MAPE and SMAPE are ranging between $0 - 1$ in our study.

7 Results analysis

In this section, we discuss about the experimental findings and also mention the inferences drawn from the findings. We discuss on each aspect of the model compared to the competitive models and the limitations on varying scenario of experimental conditions. The different experimental conditions are: (i)multi-step prediction on varying historic windows (ii) $12hr$ prediction with varying percentage of missing information in historic data (at $HWND = 24$) (iii) $12hr$ prediction at different hyper-parameters settings (at $HWND = 24$) and (iv) behaviour of model variants during multi-step prediction with varying historic windows. The obtained results are statistically significant for the given value [10, 17]. All the differences between results obtained by the proposed model and the best performing baseline are statistically significant. For HWND 24, the obtained p value difference is less than 0.000015 (RMSE, MAPE) and 0.033 (SMAPE). Next, we discuss the experimental findings in detail.

7.1 Comparison with state-of-the-art on varying historic window

The proposed model is compared to the state-of-art models for varying historic windows $hr = \{12, 24, 168\}$ i.e. when data is available for a half-day, day and a week. The multiple prediction window size are $hr = \{1, 2, 4, 6, 8, 10, 12\}$.

The results presented in Tables and Figures report a good performance of the proposed model on all the three standard benchmark metrics RMSE (Tables 2, 5 and 6), MAPE (Table 3, Fig. 5a and b), SMAPE (Table 4, Fig. 5c and d). Bold entries in Tables 2, 3, 4, 5, 6, represents the best value of that column for both the datasets. It shows the proposed model consistently report a comparatively lower error (RMSE, MAPE) in multi-step predictions, as well as the model, is capable of working on varying datasets with symmetricity in results (SMAPE). As intuitive, it can be observed that with increasing prediction window size the performance degrades for all models, though this is much less steep in the proposed model comparatively. The proposed model makes a considerable reduction in error during multi-step prediction in all cases in comparison to the baselines. A possible reason is the use of PCA$_G$ block which is an advantage over the state-of-the-art models. Hence, ST-A$_G$P has an organized and distinctive embedded features for prediction than similar models GEML, STMGCN. We report the percentage reduction in the results for the three metrics (RMSE, MAPE, SMAPE) with the best performing baseline as 37%, 36%, 9%, and 25%, 36%, 13% for two cities NYC, THS respectively at the historic window 24. This reveals that the feature aggregation in the proposed model has reached a quality performance where most of the trendy models have failed due to their constructional limitations. We observe that the generative model is not able to compete with the other baselines because such models usually require a large amount of training data. This is the reason we exclude ForGAN in graphs for better clarity in observations. Additionally, we observe that models like STMGCN, ForGAN are not performing well when cross-regions demands are predicted (like demands from region A to region B and demands from region B to region A). These models' capability limits their performance for such prediction. While GEML, TeDCaN and the proposed models are deliberately designed for such tasks. GEML, TeDCaN despite having attention mechanism, due to lack of feature organization techniques lag the proposed model in prediction performance. A key player in this case is the PCA$_G$ block. Moreover, without MTL approach and supervised attention, the proposed model can have better performance owing to its strategic feature aggregation techniques. A test snapshot of a part of the cities can be seen in Fig. 11, where one can observer correctly predicted demands of GRs (Red) by proposed model against the missed ground-truth GRs (Blue). The higher number of Red GRs than Blue GRs are self-informative.

7.2 Comparison with state-of-the-art on Missing data scenario

In this paragraph we discuss the performance of different models in missing data scenario. A percent of historic information is omitted and replaced with 0 value for data imputation to conduct this experiments. It is usually a practical scenario when either information is missing or there is noise during experimentation which corrupts the data. The experiment is conducted multiple times [11, 23] and mean results are reported. Observing the experimental results in Fig. 6a, b and c, the performance of every model seems to degrade with respect to their obtained performance (at $HWND : 24$ and $PWND : 12$) due to missing information in their training data. The proposed model is still able to maintain its lower error output than the existing state-of-art models in all three metrics. The reason is that the proposed model has a dedicated component PCA$_G$ which offers noise immunity [45, 46] by explicitly implementing

Table 2 RMSE values of Multi-step prediction at HWND 24

	Performance comparison with state-of-the-art methods PREDICTION WINDOW SIZE							
MODELS	Dataset	**1**	**2**	**4**	**6**	**8**	**10**	**12**
STMGCN	NYC	4.3216	4.6189	3.8241	4.2831	3.9912	4.1025	4.4855
	THS	4.2295	4.5792	6.7197	5.7965	4.6369	4.5318	4.7567
GEML	NYC	3.2217	3.6259	3.9061	3.8248	4.4681	3.9752	4.1194
	THS	4.7944	4.489	4.1774	4.6387	4.5888	5.0266	4.3973
ForGAN	NYC	6.0389	6.0318	6.0365	6.0204	6.0063	5.9895	5.9716
	THS	5.4897	5.7122	5.7484	5.8546	5.8468	5.8693	5.8716
	Performance comparison with variants of proposed approach							
ST-AGP∼P	NYC	2.4356	2.5847	2.8026	3.0388	3.0994	3.2798	4.4949
	THS	4.5257	4.413	4.8773	3.8589	4.5155	4.9264	**3.8247**
ST-AGP∼M	NYC	2.2432	2.5567	2.6746	2.988	3.0	3.0016	3.1328
	THS	4.0253	4.412	**3.5417**	4.8015	4.6639	5.3361	4.7033
ST-AGP∼Loss	NYC	**1.9432**	2.4207	**2.4932**	2.7647	**2.7734**	3.0852	**2.8606**
	THS	3.5694	3.1496	3.6263	5.4586	5.5066	4.8844	4.0124
ST-AGP∼PCAG	NYC	3.4524	3.7051	3.8375	4.9877	4.6037	4.1636	4.2775
	THS	4.3514	4.6272	4.6347	5.4082	6.0111	5.9079	6.1922
ST-AGP∼Cheby	NYC	2.1708	2.6831	2.6749	2.8069	2.9693	3.0882	3.0111
	THS	3.3993	3.381	3.6738	4.5076	4.6609	4.5815	4.6222
ST-AGP	NYC	1.9827	**2.2056**	2.7007	**2.7151**	2.9727	**2.7804**	2.9002
	THS	**2.7564**	**2.9964**	3.7853	**3.8584**	**3.8105**	**4.4675**	4.0364

Table 3 MAPE values of Multi-step prediction at HWND 24

	Performance comparison with state-of-the-art methods PREDICTION WINDOW SIZE							
MODELS	Dataset	**1**	**2**	**4**	**6**	**8**	**10**	**12**
STMGCN	NYC	0.1168	0.1358	0.1004	0.1334	0.1112	0.1238	0.1481
	THS	0.1125	0.1336	0.1938	0.1404	0.1162	0.1296	0.1292
GEML	NYC	0.1158	0.1281	0.1519	0.1521	.2387	0.1591	0.1584
	THS	0.1272	0.1278	0.1271	0.1448	0.1255	0.1292	0.1239
ForGAN	NYC	0.3785	0.3781	0.375	0.3728	0.3695	0.3698	0.3693
	THS	0.5619	0.5413	0.5269	0.5249	0.5239	0.5232	0.5228
	Performance comparison with variants of proposed approach							
ST-AGP∼P	NYC	0.0707	0.0732	0.0798	0.0842	0.09	0.0909	0.1232
	THS	0.1332	0.1261	0.1152	0.1005	0.0984	0.1042	0.0894
ST-AGP∼M	NYC	0.0686	0.0749	0.0829	0.0941	0.0936	0.0912	0.0924
	THS	0.0823	0.0823	0.085	0.0867	0.0868	0.0944	0.0973
ST-AGP∼Loss	NYC	0.0651	0.071	0.0771	0.0798	0.0865	0.089	0.0848
	THS	0.0789	0.081	0.0812	0.084	0.0898	**0.0899**	0.0896
ST-AGP∼PCAG	NYC	0.0917	0.0944	0.1004	0.1567	0.1207	0.1093	0.1116
	THS	0.0929	0.0911	0.0922	0.0919	0.0938	0.096	0.0989
ST-AGP∼Cheby	NYC	0.0701	0.0765	0.0825	0.0842	0.0918	0.0916	0.0862
	THS	0.0829	0.0816	0.0836	0.0872	0.0916	0.0943	0.1004
ST-AGP	NYC	0.0661	**0.0697**	**0.0765**	**0.0795**	0.0873	**0.0811**	**0.0837**
	THS	0.08	**0.0805**	0.0845	0.0854	**0.0851**	0.0905	**0.089**

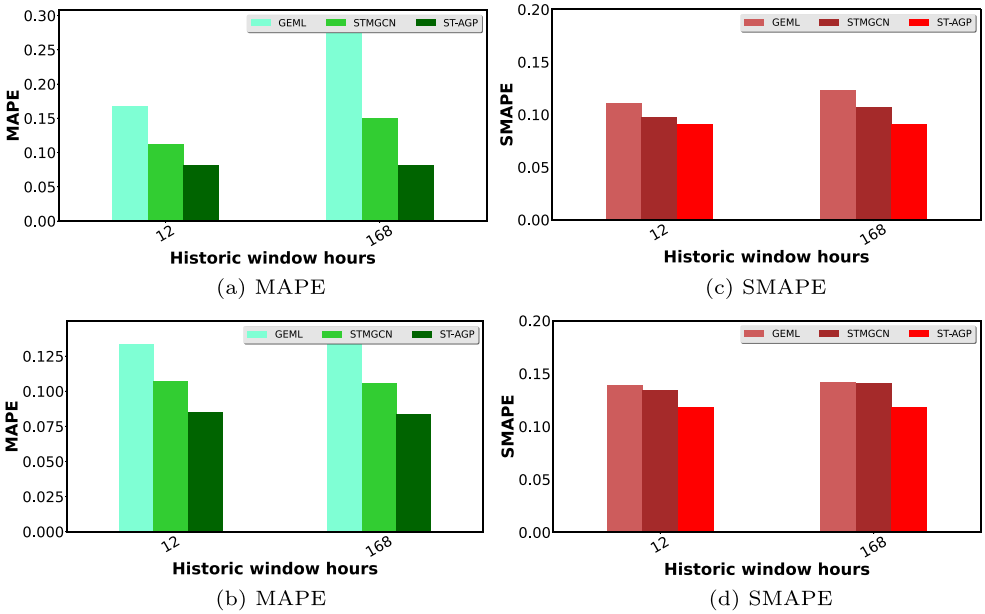


Fig. 5 Mean MAPE and SMAPE score of baselines as well as our ST-A$_G$P at HWND 12 and 168 for NYC and THS

orthogonality (Section 4). The results for all the models are showing an upward trend which is intuitive with higher portion of missing information. This is a good indication of the fact that the inclusion of PCA$_G$ module not only improves prediction but also makes the model more immune to this practical scenario. Additionally, the obtained values for NYC is lower than that of THS because of the varying temporal pattern [23] of the cities. For better clarity, we have omitted ForGAN from the results as ForGAN is having the highest error values in all and three metrics.

Table 4 SMAPE values of Multi-step prediction at HWND 24

Performance comparison with state-of-the-art methods								
PREDICTION WINDOW SIZE								
MODELS	Dataset	**1**	**2**	**4**	**6**	**8**	**10**	**12**
STMGCN	NYC	0.0999	0.1045	0.0962	0.1009	0.0963	0.0997	0.1062
	THS	0.1259	0.1312	0.1465	0.1383	0.1363	0.1367	0.1364
GEML	NYC	0.103	0.1059	0.1082	0.1098	0.1121	0.1094	0.1075
	THS	0.1367	0.1352	0.139	0.1426	0.1364	0.1403	0.1366
ForGAN	NYC	0.7446	0.7438	0.7376	0.7331	0.7265	0.7271	0.7262
	THS	1.0951	1.055	1.0265	1.0228	1.0206	1.0194	1.0184
Performance comparison with variants of proposed approach								
ST-AGP∼P	NYC	0.0891	0.0906	0.0923	0.093	0.0941	0.0934	0.1108
	THS	0.1352	0.1345	0.1358	0.1296	0.1339	0.1322	0.1205
ST-AGP∼M	NYC	0.0876	0.0901	0.0924	0.0943	0.0942	0.0947	0.0953
	THS	0.1131	0.1141	0.1179	0.1188	**0.1184**	0.1206	0.1243
ST-AGP∼Loss	NYC	**0.0864**	0.0888	**0.0899**	**0.0903**	**0.0918**	0.092	0.0932
	THS	**0.1117**	**0.1119**	**0.1166**	0.1199	0.1218	0.1229	0.122
ST-AGP∼PCAG	NYC	0.0978	0.0993	0.1001	0.1096	0.1038	0.1016	0.1018
	THS	0.1189	0.1192	0.1213	0.1234	0.1278	0.1288	0.1287
ST-AGP∼Cheby	NYC	0.0879	0.0906	0.0917	0.092	0.0934	0.0943	0.0939
	THS	0.1129	0.1142	0.1189	0.1202	0.1211	0.1246	0.123
ST-AGP	NYC	0.0873	**0.0879**	0.0907	0.0911	0.092	**0.0918**	**0.0917**
	THS	0.1123	0.1136	0.1199	**0.1179**	0.1196	**0.1206**	**0.1189**

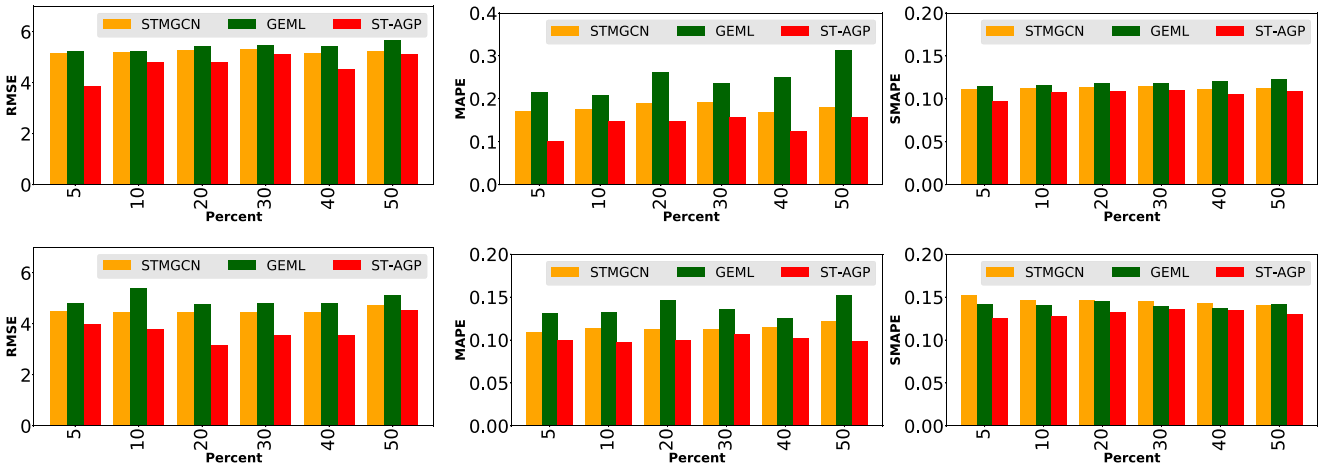


Fig. 6 RMSE, MAPE and SMAPE Score comparison of ST-A$_G$P with Baselines with varying % of missing information (at HWND:24, PWND:12) for NYC and THS

7.3 Comparison with tensor model

We discuss the result comparison of TeDCaN [23] with the proposed model by means of Figs. 7a, b and c. Due to different experimental conditions of Tensor Models ($2weeks$, $k = 70$ [23]) with a DNN model, only multi-step prediction comparison is made for $hr = 24$ with the proposed model. we can observe that TeDCaN is not able to compete for the prediction results for larger prediction windows and its performance is noticeably degrading for that. While the proposed model maintains a much slower degradation over multi-step prediction. Additionally, the historic information requirements of the proposed model are much lower than that of TeDCaN during prediction. However, as the computational complexity is different for both these models in different phases of operations i.e. training and testing, Tensor based models are less favored high prediction complexity [15]. Another observation suggests that tensor-based model can only be effective for short-term traffic forecasts. However, ST-A$_G$P performs better for both short-term and long-term predictions.

7.4 Analysing Results on varying Hyper-parameters

Observing the model's performance (at $HWND : 24$ and $PWND : 12$) on varying aggregation operation (*concatenation*, *addition*) at embedding dimension of $dim = 10$ in the result Fig. 8a b and c and those with varying embedding dimensions in Fig. 9a, b and c, we can notice that *addition* operation has better yield than concatenation. A possible reason may be that with concatenation the dimension size increases multiplicatively, introducing a more sparse representation of embedded features to affect the prediction results. However, 'addition' is able to maintain essential features in embedding space for better prediction without compromising the feature embedding dimension. As expected, both the operations have degrading results with increasing prediction window size. We notice in Fig. 9, that at lower embedding size the model performance is worst but with increasing the dimension between $10 - 40$, the results improve and become steady or worse slightly. As it is obvious that lower dimension size is not able to capture the complete feature in embedding space and very high dimension results in

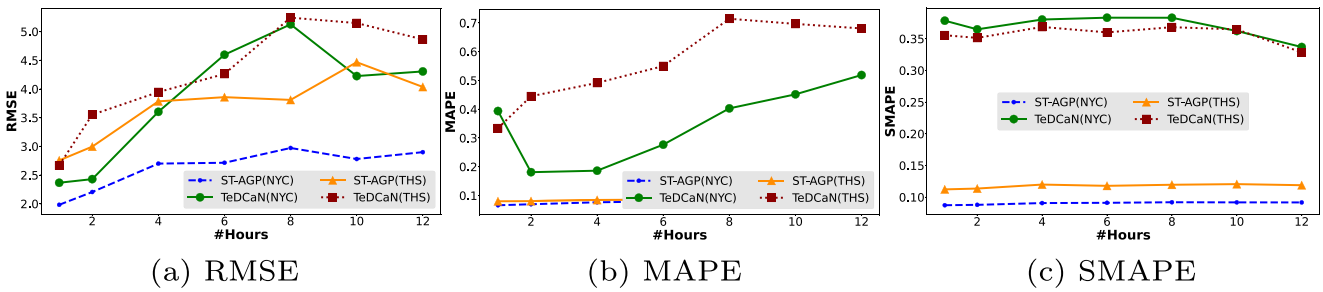


Fig. 7 RMSE, MAPE, SMAPE values of ST-A$_G$P compared to TedCaN for NYC and THS

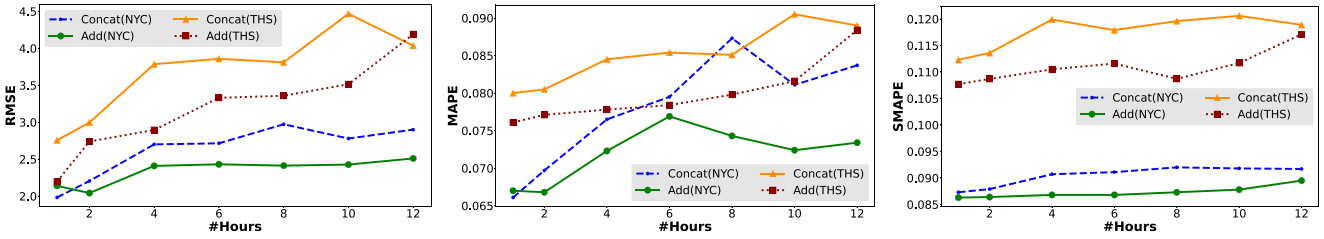


Fig. 8 RMSE, MAPE and SMAPE score comparison of ST-A$_G$P with varying aggregation operation (at HWND:24, PWND:12) for NYC and THS

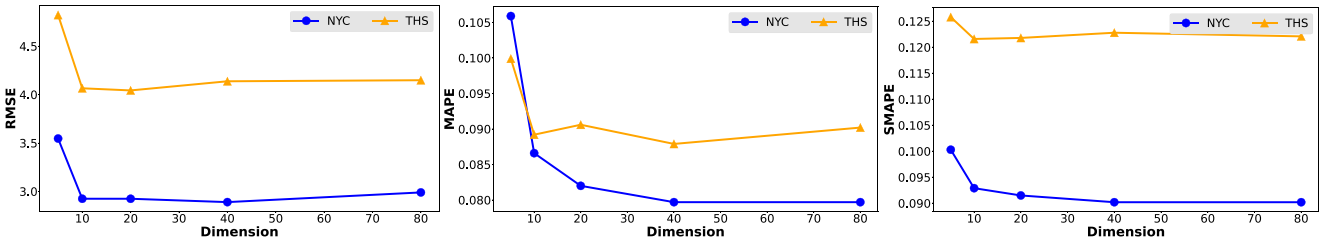


Fig. 9 RMSE, MAPE and SMAPE value of ST-A$_G$P on varying embedding dimension size for NYC and THS

Table 5 RMSE values of Multi-step prediction at HWND 12

Performance comparison with state-of-the-art methods

PREDICTION WINDOW SIZE								
MODELS	Dataset	**1**	**2 4**	**6**	**8**	**10**	**12**	
STMGCN	NYC	4.1982	4.0693	3.8454	3.7857	3.8859	4.2779	4.3914
	THS	4.1528	4.4387	4.4456	4.526	5.8002	7.01	6.5585
GEML	NYC	3.9593	3.4843	3.7983	4.3721	4.326	4.5041	4.1358
	THS	5.2944	4.4533	4.7751	5.0775	4.5514	4.5884	4.5514
ForGAN	NYC	6.0217	6.0187	6.022	5.9945	5.9945	5.989	5.9963
	THS	5.4494	5.6918	5.8754	5.6139	5.7963	5.6822	5.825
Performance comparison with variants of proposed approach								
ST-AGP∼P	NYC	2.7588	2.975	2.9912	3.257	4.8891	3.8109	4.4354
	THS	4.0674	3.9823	4.2747	3.9995	4.2601	4.2681	4.4611
ST-AGP∼M	NYC	2.3562	2.4185	2.8465	3.1269	3.0909	3.373	3.743
	THS	4.189	4.1867	4.7203	4.9295	4.4823	5.1534	5.3889
ST-AGP∼Loss	NYC	2.0767	**2.319**	2.7274	**2.8693**	3.113	3.4149	3.8807
	THS	2.9542	4.3724	**3.4889**	4.1338	**4.0912**	4.3835	**4.0167**
ST-AGP∼PCAG	NYC	3.6371	3.7832	3.9483	4.3368	5.0804	5.1724	4.8588
	THS	4.0723	3.9756	4.2107	4.244	5.3464	5.7604	5.6281
ST-AGP∼Cheby	NYC	2.4612	2.4687	2.7593	2.9784	**3.0209**	3.3744	4.0703
	THS	3.4761	3.5847	3.5719	**3.8379**	4.5455	4.1975	4.6782
ST-AGP	NYC	**1.9347**	2.4743	**2.5141**	2.9789	3.148	**3.1235**	**3.3202**
	THS	**2.6978**	**3.2087**	4.0352	4.9739	4.097	**4.0721**	4.1688

Table 6 RMSE values of Multi-step prediction at HWND 168

Performance comparison with state-of-the-art methods								
PREDICTION WINDOW SIZE								
MODELS	Dataset	**1**	**2**	**4**	**6**	**8**	**10**	**12**
STMGCN	NYC	5.3076	5.3066	5.3069	5.3095	5.3107	5.3753	3.5736
	THS	4.1606	4.3724	4.172	6.8581	7.5557	8.6172	7.5381
GEML	NYC	5.7257	5.4705	5.5568	5.6764	5.7483	4.5313	5.5312
	THS	4.6265	4.5383	5.0348	4.7697	4.8493	4.7962	4.4273
ForGAN	NYC	5.822	5.8442	5.8414	5.8321	5.8114	5.8333	5.8151
	THS	5.5088	5.6672	5.7337	5.6811	5.7476	5.8171	5.7204
Performance comparison with variants of proposed approach								
ST-AGP∼P	NYC	5.7028	5.7356	5.6756	5.5496	5.6817	5.2856	4.1526
	THS	5.3047	4.5743	4.7188	3.9693	4.229	4.4348	**4.3007**
ST-AGP∼M	NYC	2.1817	**2.1672**	2.5736	2.5956	2.9666	**2.8438**	2.9016
	THS	3.6404	5.1055	5.6136	4.7731	5.2892	4.9427	5.3981
ST-AGP∼Loss	NYC	**1.8311**	2.0798	**2.5036**	2.6257	**2.5842**	2.9333	**2.7699**
	THS	3.3999	4.7666	3.5862	3.7543	5.0268	3.8375	4.4026
ST-AGP∼PCAG	NYC	4.874	4.8714	4.7235	4.7404	4.7462	4.8740	4.8713
	THS	3.8415	4.0273	5.0973	5.4502	5.7802	6.5814	6.5937
ST-AGP∼Cheby	NYC	2.0126	2.247	2.7041	2.8959	2.9923	3.12	3.0056
	THS	**3.1319**	3.3776	3.6203	3.7842	4.6966	4.8029	4.3241
ST-AGP	NYC	1.8613	2.1912	2.5893	**2.5768**	2.8462	2.8676	2.9857
	THS	3.4783	**3.063**	**3.2886**	**3.7009**	**3.4865**	**3.8803**	4.3761

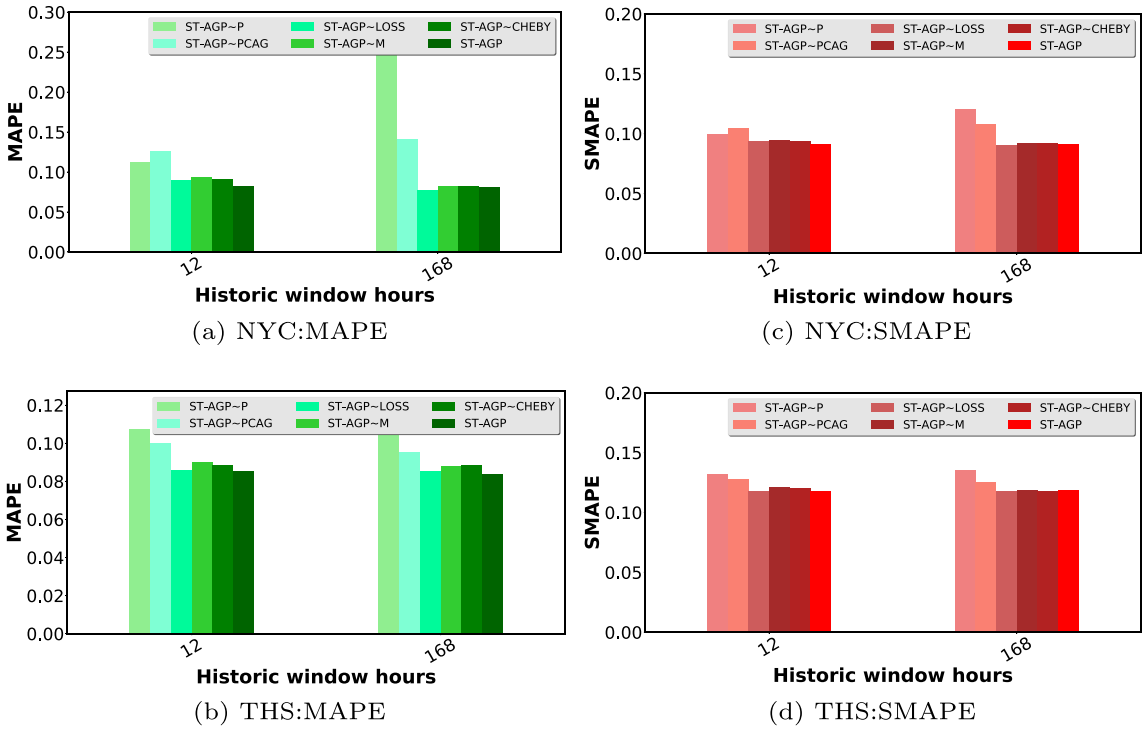


Fig. 10 Mean MAPE, SMAPE score of ST-A$_G$P and it's variants at HWND 12 and 128 for NYC and THS

(a) NYC (b) THS

Fig. 11 A part of NYC (left) and THS (right) from a test case are shown where Prediction are made against real demands. Each GR (Red) is plotted whose all demands with the connecting grids are correctly predicted at the give temporal snapshot and GR (Blue) for which the proposed model failed to make correct prediction. The maps show a good number of GRs (Red) are having correct taxi-demand prediction by ST-A$_G$P for the case

the sparse representation of feature, hence, for the current experiments on both the cities a good range of the dimension is between 10−40 and as NYC having a denser taxi-demand dataset [23] than THS, the DNN models are showing relatively better performance on it.

7.5 Ablation study

The proposed model is compared to the different variants of the model in order to observe the effectiveness of the component in the model. The effectiveness of the components like PCA$_G$, auxiliary losses ($\mathcal{L}^p_{mse}$, $\mathcal{L}^m_{mse}$), characteristics, etc. are compared and observed to be contributing as well as derogatory sometimes. The obtained results are shown in Tables 2, 5, 6 (RMSE), Table 3 (MAPE), Table 4 (SMAPE) and Fig. 10a and b (MAPE) and Fig. 10c and d (SMAPE). Observing Tables 2, 5, 6, 3, 4 that the removal of PCA$_G$ highly degrades the results of the model. Consequently, PCA$_G$ being a key component in tackling the noise and feature aggregation in multi-step prediction. Another important factors in the proposed model are $Cheby$ and $Proximity$. Absence of these from the proposed model, highly affects the results. This is intuitive since, spectral graph convolution [10, 47] is an efficient spatial aggregation approach while proximity [17] information is an important feature for traffic network. ST-A$_G$P performs relatively better than the most of the variants, sharing some of its results with ST-A$_G$P$\sim$ $\mathcal{L}_{oss}$. Indicating the presence of loss $\mathcal{L}^p_{mse}$, $\mathcal{L}^m_{mse}$ do not contribute to the model's performance but minutely worsen. Finally, we can conclude that important components (or factors) in the proposed model that greatly affects prediction results are $\mathcal{L}^p_{mse}$, $\mathcal{L}^m_{mse}$, $Cheby$ and PCA$_G$, where PCA$_G$ being the most important one owing to it's ability to create highly informative orthogonal features aggregation and noise resilient characteristics. Hence, the model and its any variant with PCA$_G$ can be a better alternative for the state-of-the-art models. A temporal snapshot of city maps from test case are shown in Fig. 11 for the prediction performance of the proposed model.

8 Conclusion and future directions

The present work introduces a deep neural network model ST-A$_G$P for the prediction of taxi-demand time-series across regions in a city for multiple steps. The proposed model is immune to the noise perturbation with a dedicated component which many recent DNN models lack. The experimental findings under various conditions show that the proposed model is considerably better than the existing predictive models in this domain. The aggregation of orthogonal spatio-temporal non-redundant features with high variance in the proposed model presents a sophisticated strategy in obtaining relevant predictive information for the task in comparison to the state-of-the-art models. Moreover, the proposed model does not rely on any additional information or datasets except an unsupervised loss. The effectiveness of the proposed model can be observed with the average performance gain achieved is $25-37\%$ over the best performing baseline model on the standard metric (RMSE) for two cities respectively. Nevertheless, the proposed model is expected to perform well on similar objective tasks like weather forecasting, traffic speed prediction, and many others applications in various domains. For investigation, we plan to conduct the experiments on many diversified datasets in those domains as our future work.

Acknowledgements This work is funded by Scheme for Promotion of Academic and Research Collaboration (SPARC) under Ministry of Human Resource Development, India, within project code P1506.

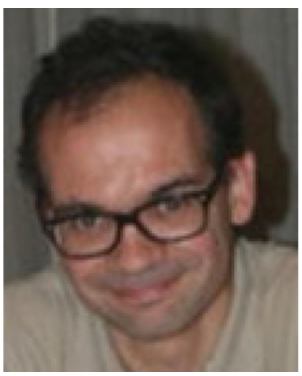
References

1. Tong Y, Chen Y, Zhou Z, Chen L, Wang J, Yang Q, Ye J, Lv W (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, p 1653–662
2. Jiang R, Song X, Fan Z, Xia T, Chen Q, Miyazawa S, Shibasaki R (2018) Deepurbanmomentum: an online deep-learning system for short-term urban mobility prediction. In: AAAI, p 784–791
3. Kuang L, Hua C, Wu J, Yin Y, Gao H (2020) Traffic volume prediction based on multi-sources gps trajectory data by temporal convolutional network. Mob Netw Appl 25(4):1405–1417
4. Kuang L, Zheng J, Li K, Gao H (2021) Intelligent traffic signalcontrol based on reinforcement learning with state reduction for smart cities. ACM Trans Int Technol (TOIT) 21(4):1–24
5. Huang X, Ye Y, Wang C, Yang X, Xiong L (2021) A multi-mode traffic flow prediction method with clustering based attention convolution lstm. Applied Intelligence, p 1–14
6. Zhang Y, Yang Y, Zhou W, Wang H, Ouyang X (2021) Multi-city traffic flow forecasting via multi-task learning. Applied Intelligence, p 1–19
7. Tu Y, Lin S, Qiao J, Liu B (2021) Deep traffic congestion prediction model based on road segment grouping. Applied Intelligence, p 1–23
8. Chiabaut N, Faitout R (2021) Traffic congestion and travel time prediction based on historical congestion maps and identification of consensual days. Transp Res Part C Emerg Technol 124:102920. https://doi.org/10.1016/j.trc.2020.102920
9. Geng X, Li Y, Wang L, Zhang L, Yang Q, Ye J, Liu Y (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, p 3656–3663
10. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks:, A deep learning framework for traffic forecasting. arXiv:1709.04875
11. Tan H, Wu Y, Shen B, Jin PJ, Ran B (2016) Short-term traffic prediction based on dynamic tensor completion. IEEE Trans Intell Transp Syst 17(8):2123–2133
12. Lee S, Fambro DB (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transp Res Rec 1678(1):179–188
13. Kumar SV, Vanajakshi L (2015) Short-term traffic flow prediction using seasonal arima model with limited input data. Eur Transp Res Rev 7(3):21
14. Chandra SR, Al-Deek H (2009) Predictions of freeway traffic speeds and volumes using vector autoregressive models. J Intell Transp Syst 13(2):53–72
15. Ren J, Xie Q (2017) Efficient od trip matrix prediction based on tensor decomposition. In: Mobile Data Management (MDM), 2017 18th IEEE International Conference On, p 180–185. IEEE
16. Bhanu M, Mendes-Moreira J, Chandra J (2020) Embedding traffic network characteristics using tensor for improved traffic prediction. IEEE Transactions on Intelligent Transportation Systems, p 1–13
17. Wang Y, Yin H, Chen H, Wo T, Xu J, Zheng K (2019) Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1227–1235
18. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. IJCAI, p 3634–3640
19. Zhou X, Shen Y, Zhu Y, Huang L (2018) Predicting multi-step citywide passenger demands using attention-based neural networks. In: Proceedings of the Eleventh ACM international conference on web search and data mining, p 736–744
20. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2020) T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Trans Intell Trans Syst 21(9):3848–3858. https://doi.org/10.1109/TITS.2019.2935152
21. Liu P, Qiu X, Huang X (2017) Adversarial multi-task learning for text classification. In: Proceedings of the 55th annual meeting of the association for computational linguistics
22. Priya S, Upadhyaya A, Bhanu M, Kumar Dandapat S, Chandra J (2020) Endea: Ensemble based decoupled adversarial learning for identifying infrastructure damage during disasters. In: Proceedings of the 29th ACM international conference on information & knowledge management, p 1245–1254
23. Bhanu M, Mendes-Moreira J, Chandra J (2020) Embedding traffic network characteristics using tensor for improved traffic prediction IEEE Transactions on Intelligent Transportation Systems
24. Yao H, Liu Y, Wei Y, Tang X, Li Z (2019) Learning from multiple cities: a meta-learning approach for spatial-temporal prediction. In: The world wide web conference, p 2181–2191
25. Kuang L, Gong T, OuYang S, Gao H, Deng S (2020) Offloading decision methods for multiple users with structured tasks in edge computing for smart cities. Futur Gener Comput Syst 105:717–729
26. Liu Y, Liu C, Lu X, Teng M, Zhu H, Xiong H (2017) Point-of-interest demand modeling with human mobility patterns. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, p 947–955. ACM
27. Moreira-Matias L, Gama J, Ferreira M, Mendes-Moreira J, Damas L (2013) Predicting taxi–passenger demand using streaming data. IEEE Trans Intell Transp Syst 14(3):1393–1402
28. Beiraghi M, Ranjbar A (2011) Discrete fourier transform based approach to forecast monthly peak load. In: 2011 Asia-Pacific power and energy engineering conference, p 1–5. IEEE
29. Lee S, Fambro D (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. In: Transportation research record:, journal of the transportation research board (1678), p 179–188
30. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
31. Kumar SV, Vanajakshi L (2015) Short-term traffic flow prediction using seasonal arima model with limited input data. Eur Trans Res Rev 7(3):21
32. Trasarti R, Guidotti R, Monreale A, Giannotti F (2017) Myway: Location prediction via mobility profiling. Inf Syst 64:350–367
33. Hoang MX, Zheng Y, Singh AK (2016) Fccf: forecasting citywide crowd flows based on big data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems, p 1–10
34. Tan H, Feng G, Feng J, Wang W, Zhang YJ, Li F (2013) A tensor-based method for missing traffic data completion. Trans Res Part C:, Emerg Technol 28:15–27
35. Liu J, Musialski P, Wonka P, Ye J (2012) Tensor completion for estimating missing values in visual data. IEEE Trans Pattern Anal Mach Intell 35(1):208–220

36. Niesing J (1997) Simultaneous Component and Factor Analysis Methods for Two or More groups: a Comparative Study vol 1997. DSWO Press, Leiden University leiden: The Netherlands ???
37. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI conference on artificial intelligence
38. Deng D, Shahabi C, Demiryurek U, Zhu L, Yu R, Liu Y (2016) Latent space model for road networks to predict time-varying traffic. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, p 1525–1534
39. Zhu Y, Zhang W, Chen Y, Gao H (2019) A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment. EURASIP J Wirel Commun Netw 2019(1):1–18
40. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, p 1024–1034
41. Seo Y, Defferrard M, Vandergheynst P, Bresson X (2018) Structured sequence modeling with graph convolutional recurrent networks. In: International conference on neural information processing, p 362–373. Springer
42. Lin K, Xu X, Gao H (2021) Tscrnn: a novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of iiot. Comput Netw 190:107974
43. Koochali A, Schichtel P, Dengel A, Ahmed S (2019) Probabilistic forecasting of sensory data with generative adversarial networks–forgan. IEEE Access 7:63868–63880
44. Zheng C, Fan X, Wang C, Qi J (2020) Gman: a graph multi-attention network for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 1234–1241
45. Householder AS (1958) Unitary triangularization of a nonsymmetric matrix. J ACM (JACM) 5(4):339–342
46. Stewart GW (1980) The efficient generation of random orthogonal matrices with an application to condition estimators. SIAM J Numer Anal 17(3):403–409
47. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv:1609.02907
48. Azzouni A, Pujolle G (2017) A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv:1705.05690
49. Toqué F, Côme E, El Mahrsi MK, Oukhellou L (2016) Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC), p 1071–1076. IEEE
50. Bhanu M, Priya S, Dandapat SK, Chandra J, Mendes-Moreira J (2018) Forecasting traffic flow in big cities using modified tucker decomposition. In: International conference on advanced data mining and applications, p 119–128. Springer
51. Bhanu M, Chandra J, Mendes-Moreira J (2018) Enhancing traffic model of big cities: Network skeleton & reciprocity. In: Communication Systems & Networks (COMSNETS), 2018 10th International Conference On, p 121–128. IEEE
52. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2019) t-gcn: A temporal graph convolutional network for traffic prediction IEEE Transactions on Intelligent Transportation Systems
53. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: Thirty-second AAAI conference on artificial intelligence

Manish Bhanu is a research scholar student in the department of Computer Science and Engineering at Indian Institute of Technology, Patna. He has worked with LIAAD, INESC TEC, Porto, Portugal. His current research interest includes Intelligent Transportation Systems, Data mining and Text classification and Information retrieval. His research domain also focuses on apply deep learning and machine learning techniques for traffic flow prediction, text classification during disaster and rumor identification.

Shalini Priya is currently working as a postdoctoral research associate in Oak Ridge National Laboratory, USA. She has completed her PhD in Computer Science and Engineering from Indian Institute of Technology, Patna. Her current research interest includes Machine leaning, Deep Learning, Natural Language Processing, Data mining, Text classification, Time-series Prediction, and Information Retrieval. Major application areas are Disaster and Emergency, Computational Journalism, Intelligent transportation system and health sciences.



João Mendes-Moreira received the Ph.D. degree in engineering sciences from the University of Porto, Porto, Portugal. He is an Assistant Professor with the Department of Informatics Engineering, Faculty of Engineering, University of Porto. He is also a Researcher with the Laboratory for Artificial Intelligence and Decision Support, at INESC TEC, Porto, Portugal. His research works focus on applied machine learning, supervised learning and intelligent transportation systems.

Joydeep Chandra is an Assistant Professor in the Department of Computer Sc. and Eng. at Indian Institute of Technology, Patna, India. He received his PhD from IIT, Kharagpur, India in 2012. He was also a research fellow at the Chair of Systems Design at ETH Zurich. His research interest includes modeling of social networks, studying diffusion of information and identifying influentials. Application domains include Journalism, Disaster, Healthcare, Intelligent transportation system and Crimes on the Web.