Interpretable Spatial-Temporal Attention Graph Convolution Network for Service Part Hierarchical Demand Forecast

Wenli Ouyang[1], Yahong Zhang[1], Mingda Zhu[1], Xiuling Zhang[1], Hongye Chen[1], Yinghao Ren[1], Wei Fan[1]

[1]Artificial Intelligence Lab, Lenovo Research, Beijing, China
{ouyangwl1,zhangyh33,zhumd4,zhangxl35,chenhy17,renyh6,fanwei2}@lenovo.com

Abstract. Accurate service part demand forecast plays a key role in service supply chain management. It enables better decision making in the planning of service part procurement and distribution. To achieve high responsiveness, the service supply chain network exhibits a hierarchical structure: forward stocking locations (FSL) close to the end customer, distribution centers (DC) in the middle and center hub (CH) at the top. Hierarchical forecasts require not only good prediction accuracy at each level of the service supply chain network, but also the consistency between different levels. The accuracy and consistency of hierarchical forecasts are important to be interpretable to the decision-makers (DM). Moreover, service part demand data is the spatial-temporal time series that the observations made at neighboring regions and adjacent timestamps are not independent but dynamically correlated with each other. Recent advances in deep learning enable promising results in modeling the complex spatial-temporal relationship. Researchers use convolutional neural networks (CNN) to model spatial correlations and recurrent neural networks (RNN) to model temporal correlations. However, these deep learning models are non-transparent to the DMs who broadly require justifications in the decision-making processes. Here an interpretable solution is in the urgent demand. In this paper, we present an interpretable general framework **STAH** (**S**patial-**T**emporal **A**ttention Graph Convolution network for **H**ierarchical demand forecast). We evaluate our approach on Lenovo Group Ltd.'s service part demand data in India. Experimental results demonstrate the efficacy of our approach, showing superior accuracy while increasing model interpretability.

1 Introduction

Service parts for products like notebooks, cellphones, household appliances, and automobiles have grown into a business worth more than $200 billion worldwide [1]. Service parts need to be managed at an appropriate level within the service supply chain to provide after-sales services to customers. Considering the high number of parts managed, the high responsiveness required due to downtime cost for customers and the risk of stock obsolescence, the service supply chain management is a difficult task for decision-makers (DM). The most difficult one in the decision processes is how to estimate service part demand accurately. Generally, the service supply chain network is multiple hierarchical structures: forward stocking locations (FSL) close to the end customer, distribution centers (DC) in the middle and center hub (CH) at the top. A good service part management needs an accurate demand forecast at each level and the consistent demand forecast between different levels. For instance, service part demand in the CH can be disaggregated into one of DCs, which are further disaggregated into one of FSLs. The aggregation consistency is a critical point for decision-maker (DM) to interpret and accept the forecast results, which means the disaggregated demands should add up equally to the aggregated ones. Moreover, service part demand data is spatial-temporal time series that the observations made at neighboring regions and timestamps are not independent but dynamically correlated with each other. The key challenge to providing accurate service part demand forecast is how to discover inherent spatial-temporal patterns and extract the spatial-temporal correlation of data effectively. In recent years, many researchers use deep learning methods to deal with spatial-temporal data, i.e., convolutional neural network (CNN) to extract spatial features of grid-based data and recurrent neural network (RNN) to extract temporal features of time-series data. Compared with the time series analysis model and traditional machine learning method, the deep learning method achieves great results and shows its advantages in modeling end-to-end nonlinear interactions, incorporating exogenous variable and extract features automatically [2]. However, these deep learning models are described as “black-box” and non-transparent to DMs who broadly require justifications in the decision-making processes.

To tackle the above challenges, we propose an interpretable general framework **STAH** (**S**patial-**T**emporal **A**ttention Graph Convolution network for **H**ierarchical demand forecast) to predict service part demand hierarchically. Instead of using CNNs, this model uses interpretable hierarchical graph convolution networks (GCN). It is capable to handle non-Euclidean hierarchical data structure such as the service supply chain network structure. To increase interpretability even further, attention mechanism is used in both hierarchical GCNs and RNN encoder-decoder to localize discriminative regions and timestamps both spatially and temporally. The main contributions of this paper are summarized as follows:

- We develop a neural network structure for the hierarchical forecast that met the aggregation consistency inherently. The neural network has multiple levels of outputs, each of which is corresponding to each level of the service supply chain network. The high-level output is the sum of the connected low-level ones. The objective function of this model is the combination of the objective function of each output at each level.
- We propose a spatial hierarchical attention module that captures multilevel spatial correlations from the graph-based hierarchical service supply chain network and a temporal alignment attention module that identify the most relevant historical observations and align forecast results with them.
- We apply inter-temporal regularization to restrict the difference of the learned spatial attention maps among different timestamps. This can help to avoid the case in which

the learned attention maps of each spatial region focus on one specific temporal state and largely ignore the other temporal ones.

2 Related Review

2.1 Hierarchical forecast

Some time series analysis models such as ARIMA (Autoregressive Integrated Moving Average model), ETS (Smoothing State Space model), etc. are applied in hierarchical forecast [3–6]. This forecasting method estimates time series at all levels independently. This approach doesn't guarantee aggregation consistency in the hierarchical structure and the separate predictive models don't take account of spatial correlations between each region. The "bottom-up" approach is adopted to meet the aggregation consistency constraint [5]. It forecasts all of the bottom-level disaggregated series and then adds the results of the forecast to form the higher-level series until it reaches the top-level one. However, this approach still doesn't consider spatial correlations and the disaggregated data tends to have a low signal-to-noise ratio, the overall prediction accuracy will be poor [7]. The optimal combined forecasting is the mainstream [4, 5, 8]. It estimates the initial forecast at bottom-levels and reconciles these forecasts based on aggregation consistency. Ordinary Least Square (OLS) and Weight Least Square (WLS) are used to estimate the covariance matrix based on historical observations.

2.2 Convolutions on graphs

CNNs can effectively extract the local patterns of the standard grid data. To generalize CNNs to data of graph structures, two basic approaches are proposed. One is to perform convolutional filtering on graph's nodes and their neighbors directly [9], the other is to manipulate in the spectral domain with graph Fourier transforms [10]. However, this method requires explicitly computing the Laplacian eigenvectors, which is impractical for real large graphs. [11] find a model to circumvent this problem by using Chebyshev polynomial approximation to realize eigenvalue decomposition. [12] simply this model by limiting the application of each filter to the 1-neighbor of each node, approximating the largest eigenvalue and applying a normalization trick to the convolution matrix. In this way, they reduce the computational complexity to linear. This simplified model is called GCN, which is used in this paper.

2.3 Attention mechanism

Since the attention mechanism is propose by [13], it has been applied in various tasks such as natural language processing, image caption and speech recognition. The attention mechanism can select the information that is relatively critical to the current task from all inputs. Together with RNNs and CNNs, attention mechanism has proven to be useful to learn representation and improve performances in applied tasks [14, 15]. Recently, [16] extend the attention mechanism to process graph-structured data and achieved state-of-art results. In the time series forecast task, [17] proposed a multi-level attention network to adjust the correlations among multiple time series generated from different locations. [18] proposed a spatial-temporal forecast model that applies the attention mechanism in both spatial and temporal dimensions.

3 STAH: the model

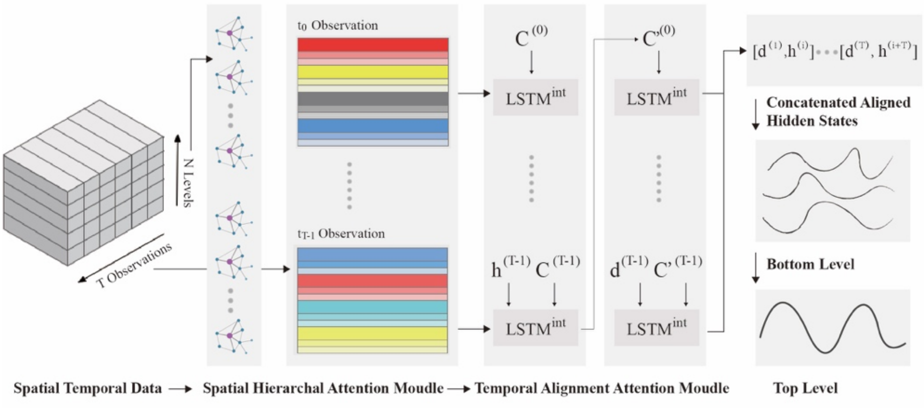

Figure 1. The system architecture of the proposed **S**patial-**T**emporal **A**ttention Graph Convolution network for **H**ierarchical demand forecast (STAH).

In this section, we first mathematically formulate the definition of hierarchical service part demand forecast and then we present the technical details of the proposed model STAH. The system architecture of the proposed model STAH is shown in Figure 1. We represent the output layer in a hierarchical structure that has multiple levels of outputs, each of which is corresponding to each level of the service supply chain network. The high-level output is the sum of the connected low-level ones. The aggregation consistency is met inherently in the hierarchical output layer. The proposed neural network STAH is composed of spatial hierarchical attention module and temporal alignment attention module. The values of all features at each timestamp are firstly processed by spatial hierarchical attention module and then feed into temporal alignment attention module to generate hierarchical forecast.

3.1 Hierarchical Service Part Demand Forecast

The service supply chain network is a hierarchical structure that have multiple levels, as shown in Figure 2. Suppose service part demand and exogenous variables are recorded as time series on each region at each level as $\boldsymbol{x}_{i,j}^{(t)} = (y_{i,j}^{(t)}, a_{i,j}^{(t)}, b_{i,j}^{(t)}, \ldots, c_{i,j}^{(t)}) \in \mathbb{R}^{F+1}$, where y is service part demand. $a, b, \ldots, c$ are exogenous variables. i, j and t represent level i, region j and timestamp t. F is the number of exogenous variables. $\boldsymbol{X}_i^{(t)} = (\boldsymbol{x}_{i,1}^{(t)}, \boldsymbol{x}_{i,2}^{(t)}, \ldots, \boldsymbol{x}_{i,n_i}^{(t)}) \in \mathbb{R}^{n_i \times (F+1)}$ denotes the values of all the features of all regions at level i and time t. We define the number of the regions at all levels as $n = \sum_1^N n_i$. $\boldsymbol{X}^{(t)} = (\boldsymbol{X}_1^{(t)}, \boldsymbol{X}_2^{(t)}, .., \boldsymbol{X}_N^{(t)}) \in \mathbb{R}^{n \times (F+1)}$ denotes the value of all the features of all the regions at all the level at time t. The same processes apply to y and $\boldsymbol{Y}_i^{(t)} \in \mathbb{R}^{n_i}$ denotes the service part demand of all regions at level i and time t. $\boldsymbol{Y}^{(t)} \in \mathbb{R}^n$ denotes the service part demand of all the regions at all the level at time t. Then the hierarchical service part demand forecast problem is formulated as a multi-step prediction given input with a fixed temporal length, i.e., learning a function $f\colon \mathbb{R}^{n \times (F+1) \times T} \rightarrow \mathbb{R}^{n \times \Delta}$ that maps the historical values of all the features to the demand in the following interval Δ.

$$\left[\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(T)}\right] \xrightarrow{f(\cdot)} \left[\boldsymbol{Y}^{(T+1)}, \dots, \boldsymbol{Y}^{(T+\Delta)}\right] \tag{1}$$

The aggregation consistency can be formulated as $\boldsymbol{Y}^{(t)} = T_c \boldsymbol{Y}_1^{(t)}$, where index 1 represents the bottom level and T_c denotes an $n \times n_1$ summing matrix derived from the hierarchical structure. It consists of an $(n - n_1) \times n_1$ submatrix $T_{c,a}$ and an $n_1 \times n_1$ identity matrix.

$$T_c = \begin{bmatrix} T_{c,a} \\ I_{n_1} \end{bmatrix} \tag{2}$$

3.2 Spatial Hierarchical Attention Module

The service supply chain network generally organizes as a hierarchical graph structure. In order to process this multilevel non-Euclidian data structure, multiple GCNs are used, each of which is applied to process the data of each level in the hierarchical graph structure, as shown in Figure 2. We introduce the notion of graph convolution operator "$* \mathcal{G}$" based on the conception of spectral graph convolution, as the multiplication of $\boldsymbol{X}_i^{(t)}$ at level i and time t with a kernel Θ,

$$\Theta * \mathcal{G}\, \boldsymbol{X}_i^{(t)} = \Theta(L)\boldsymbol{X}_i^{(t)} = \Theta(U\Lambda U^T)\boldsymbol{X}_i^{(t)} = U\Theta(\Lambda)U^T\boldsymbol{X}_i^{(t)} \tag{3}$$

where graph Fourier basis $U \in \mathbb{R}^{n_i \times n_i}$ is the matrix of eigenvectors of the normalized graph Laplacian $L = I_{n_i} - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = U\Lambda U^T \in \mathbb{R}^{n_i \times n_i}$. $D \in \mathbb{R}^{n_i \times n_i}$ is the diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$. $\Lambda \in \mathbb{R}^{n_i \times n_i}$ is the diagonal matrix of eigenvalues of L. Two approximation strategies are applied to simplify equation (1). One approximation is Chebyshev Polynomial Approximation. The kernel Θ can be restricted to a polynomial of Λ as $\Theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$, where $\theta \in \mathbb{R}^K$ is a vector of polynomial coefficients. K is the kernel size of graph convolution. Chebyshev polynomial $T_k(x)$ is used to approximate kernels as a truncated expansion of order $K-1$ as $\Theta(L) \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})$ with rescaled $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_n$, where λ_{max} denotes the largest eigenvalue of L [19]. The graph convolution can then be rewritten as,

$$\Theta * \mathcal{G}\, \boldsymbol{X}_i^{(t)} = \Theta(L)\boldsymbol{X}_i^{(t)} \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})\boldsymbol{X}_i^{(t)} \tag{4}$$

where $T_k(\tilde{L}) \in \mathbb{R}^{n_i \times n_i}$. Another approximation is 1st-order Approximation. Set $K = 1$ and assume $\lambda_{max} \approx 2$. Thus, the equation (4) can be simplified to,

$$\Theta * g\, \boldsymbol{X}_i^{(t)} \approx \theta_0 \boldsymbol{X}_i^{(t)} - \theta_1 (D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \boldsymbol{X}_i^{(t)} \tag{5}$$

where θ_0 and θ_1 are two shared parameters of the kernel. Set $\theta = \theta_0 = -\theta_1$ and renormalize W and D by $\widetilde{W} = W + I_{n_i}$ and $\widetilde{D}_{ii} = \sum_j \widetilde{W}_{ij}$. Then, the graph convolution can be expressed as,

$$\Theta * g\, \boldsymbol{X}_i^{(t)} = \theta(\widetilde{D}^{-\frac{1}{2}} \widetilde{W} \widetilde{D}^{-\frac{1}{2}}) \boldsymbol{X}_i^{(t)} \tag{6}$$

Applying a stack of graph convolutions with the 1st-order approximation vertically can achieves the similar effect as K-localized convolutions do horizontally. In the spatial dimension, the service parts demand of different regions has highly dynamic influence among each other. Here, we propose a hierarchical attention model to capture the spatial correlations at the same level in the hierarchical graph structure. The spatial attention at level i and time t is defined as,

$$\boldsymbol{S}_i^{(t)} = \boldsymbol{V} \cdot \sigma(\boldsymbol{X}_i^{(t)} \boldsymbol{W} (\boldsymbol{X}_i^{(t)})^T + \boldsymbol{b}) \tag{7}$$

$$\tilde{S}_{i,j,k}^{(t)} = \frac{\exp(S_{i,j,k}^{(t)})}{\sum_{k=1}^{N} \exp(S_{i,j,k}^{(t)})} \tag{8}$$

where $\boldsymbol{S}_i^{(t)} \in \mathbb{R}^{n_i \times n_i}$ represents attention matrix at level i and time t and $S_{i,j,k}^{(t)}$ is an element in $\boldsymbol{S}_i^{(t)}$ representing the correlation strength between region j and region k. $\boldsymbol{W} \in \mathbb{R}^{(F+1)\times(F+1)}$, $\boldsymbol{V}$, $\boldsymbol{b} \in \mathbb{R}^{n_i \times n_i}$ are learnable parameters and sigmoid σ is used as the activation function. When performing the graph convolution, the spatial attention matrix $\tilde{\boldsymbol{S}}_i^{(t)} \in \mathbb{R}^{n_i \times n_i}$ is accompanied with the $T_k(\tilde{L})$. The graph convolution formula (4) changes to

$$\Theta * g\, \boldsymbol{X}_i^{(t)} = \Theta(L) \boldsymbol{X}_i^{(t)} \approx \sum_{k=0}^{K-1} \theta_k (T_k(\tilde{L}) \odot \tilde{\boldsymbol{S}}_i^{(t)}) \boldsymbol{X}_i^{(t)} \tag{9}$$

And the formula (6) changes to

$$\Theta * g\, \boldsymbol{X}_i^{(t)} = \theta((\widetilde{D}^{-\frac{1}{2}} \widetilde{W} \widetilde{D}^{-\frac{1}{2}}) \odot \tilde{\boldsymbol{S}}_i^{(t)}) \boldsymbol{X}_i^{(t)} \tag{10}$$

where $\odot$ is the Hadamard product. In order to encourage the spatial attention model to preserve the similarity and meanwhile avoid focusing on one timestamp, we design the inter-temporal regularization that measures the difference among spatial attention matrix. We employ the square Frobenius Norm of the difference between $\boldsymbol{S}_i^{(t_1)}$ and $\boldsymbol{S}_i^{(t_2)}$, defined as

$$Reg = \left\| \boldsymbol{S}_i^{(t_1)} - \boldsymbol{S}_i^{(t_2)} \right\|_F = \sqrt{\sum_{j=1}^{n_i} \sum_{k=1}^{n_i} (S_{i,j,k}^{(t_1)} - S_{i,j,k}^{(t_2)})^2} \tag{11}$$

We randomly choose m pairs of spatial attention matrixes from each training sample for this regularization term and add this term Reg to the original objective function for the model training.

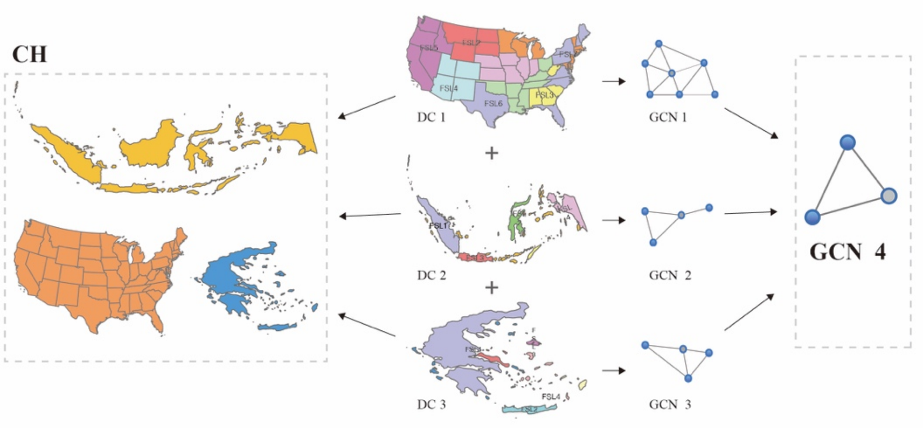

Figure 2. The hierachical structure of a typical service supply chain network that consists one CH, three DCs and many FSLs. Four GCNs are used to model spatial correlation within one CH and three DCs.

3.3 Temporal Alignment Attention Module

We denote $\widetilde{\boldsymbol{X}}_i^{(t)} = \Theta * g\, \boldsymbol{X}_i^{(t)} \in \mathbb{R}^{n_i \times D}$ and $\widetilde{\boldsymbol{X}}^{(t)} = (\widetilde{\boldsymbol{X}}_1^{(t)}, \widetilde{\boldsymbol{X}}_2^{(t)}, .., \widetilde{\boldsymbol{X}}_N^{(t)}) \in \mathbb{R}^{n \times D}$. D is feature dimension after the graph convolution. Take $\{\widetilde{\boldsymbol{X}}^{(t)}\}_{t=1}^{T}$ as input, RNN encodes $\{\widetilde{\boldsymbol{X}}^{(t)}\}_{t=1}^{T}$ into hidden states $\{\boldsymbol{h}^{(t)}\}_{t=1}^{T}$ via:

$$\boldsymbol{h}^{(t)} = LSTM^{int}(\widetilde{\boldsymbol{X}}^{(t)}, \boldsymbol{h}^{(t-1)}) \tag{12}$$

where $LSTM^{int}$ is a long short memory architecture (LSTM) encoder to capture the long-range dependency proposed by [20]. To predict the desired service part demands $\{\boldsymbol{Y}^{(t)}\}_{t=T+1}^{T+\Delta}$, we adopt a LSTM decoder defined as

$$\boldsymbol{d}^{(t)} = LSTM^{dec}(\widetilde{\boldsymbol{X}}^{(t)}, \boldsymbol{d}^{(t-1)}) \tag{13}$$

where $\boldsymbol{d}^{(t)} \in \{\boldsymbol{d}^{(t)}\}_{t=T+1}^{T+\Delta}$ is the hidden state to learn in the decoding process. Ideally, the learned hidden states $\{\boldsymbol{h}^{(t)}\}_{t=1}^{T}$ and $\{\boldsymbol{d}^{(t)}\}_{t=T+1}^{T+\Delta}$ carries contextual information in current and previous timestamps. However, the performance of the encoder-decoder networks decreases significantly when the length of time series increases. To alleviate this problem, we propose a temporal alignment attention model. At first, we concatenate Δ successive encoder hidden states as:

$$\boldsymbol{P}_i = [\boldsymbol{h}^{(i)}; \boldsymbol{h}^{(i+1)}; \dots; \boldsymbol{h}^{(i+\Delta-1)}],\ 1 \leq\ i \leq T - \Delta + 1 \tag{14}$$

Similarly, we concatenate all the decoder hidden states as:

$$\widetilde{\boldsymbol{P}} = \left[\boldsymbol{d}^{(T+1)}; \boldsymbol{d}^{(T+2)}; \dots ; \boldsymbol{d}^{(T+\Delta)}\right] \tag{15}$$

Then, we compute the relevance score between $\boldsymbol{P}_i \in \{\boldsymbol{P}_i\}_{i=1}^{T-\Delta+1}$ and $\widetilde{\boldsymbol{P}}$ as $e_i = \widetilde{\boldsymbol{P}}\boldsymbol{P}_i{}^T$ and find the maximum one, with $i_{max} = \text{argmax}\{e_i\}_{i=1}^{T-\Delta+1}$.Finally, we merge each pair in $\{\boldsymbol{h}^{(t)}\}_{t=i_{max}}^{i_{max}+\Delta-1}$ and $\{\boldsymbol{d}^{(t)}\}_{t=T+1}^{T+\Delta}$ into $\{\widetilde{\boldsymbol{d}}^{(t)}\}_{t=T+1}^{T+\Delta}$ that contains the aligned long-distance encoder hidden states. We approximate the future service part demands with regression:

$$\widehat{\boldsymbol{Y}}_1^{(t)} = \boldsymbol{A}\widetilde{\boldsymbol{d}}^{(t)} + \boldsymbol{B} \tag{16}$$

$$\widehat{\boldsymbol{Y}}^{(t)} = T_c \boldsymbol{Y}_1^{(t)} \tag{17}$$

where $\widehat{\boldsymbol{Y}}_1^{(t)}$ denotes the predicted service part demand at bottom level and $\widehat{\boldsymbol{Y}}^{(t)}$ denotes the predicted service part demands at all levels of the hierarchical structure. $\boldsymbol{A}$ and $\boldsymbol{B}$ are parameters to learn. For model learning, we apply mean squared error coupled with regularization term Reg multipled by a coefficient λ:

$$\mathcal{L}_{loss} = \frac{1}{N}\left(\sum_{n=1}^{N}\left(\sum_{t=T+1}^{T+\Delta}\left(\widehat{\boldsymbol{Y}}^{(t)} - \boldsymbol{Y}^{(t)}\right) + \lambda\, Reg\right)\right) \tag{18}$$

where *N* is the number of the batch size. In the training procedure, we leverage mini-batch Stochastic Gradient Decent (SGD) based algorithm, named Adam. We set the batch size as 128 and the starting learning rate as 0.001 which is reduced by 10% after 10,000 iterations.

4 Experiments

4.1 Experimental Datasets

We use one real-life dataset: Lenovo Group Ltd.'s service part demand data in India that has one DC and 17 FSLs. The dataset contains 17,467 stock keeping units (SKU) of service parts demand over five years. Except service parts demand data, other internal data like installed base, service parts category, etc., and external data like weather condition and holiday are collected. All the data is aggregated by week. We reduce the original dimensionality of categorical data by taking the 4th root of the number of categories. We use four years of data as the training set, the following six months as the validation set, and the final six months as the testing set.

4.2 Experimental Setup

In the experiments, we compute 26 weeks ahead rolling forecasts with 52 weeks historical demand observations and other features, i.e. $\Delta = 26$ and $T = 52$. The Adjacency matrix of the DC graph in India is computed based on the distances among FSLs in the service supply chain network. The weighted adjacency matrix $\boldsymbol{W}$ can be formed as,

$$w_{i,j}=\begin{cases}\exp\left(-\frac{d_{i,j}^2}{\sigma^2}\right), i\neq j \text{ and } \exp\left(-\frac{d_{i,j}^2}{\sigma^2}\right)\geq\epsilon\\ 0, \text{otherwise.}\end{cases}$$

where $w_{i,j}$ is the weight of edge which is decided by $d_{i,j}$ (the distance between FSL i and j). σ^2 and ϵ are thresholds to control the matrix $\boldsymbol{W}$, assigned to $500km$ and $10km$, respectively. Considering the computing efficiency, 1st-order approximation is used in this paper. Two stacks of the 1st-order approximation GCNs are applied vertically and convolution kernel of the first and the second GCN layers are $D_1 = 64$ and $D_2 = 32$, respectively. The LSTM encoder-decoder network is also two-layer LSTM, with 512 and 128 hidden states, respectively. The number of pairs of spatial attention matrixes $m = 10$ in the term Reg. To measure and evaluate the performance of different methods, Mean Absolute Percentage Errors (MAPE), and Root Mean Squared Errors (RMSE) are adopted. We compare our framework STAH with the following baselines: 1) Bottom-up Moving Average (MA); 2) Bottom-up Auto-Regressive Integrated Moving Average (ARIMA); 3) LSTM encoder-decoder (STAH without the spatial hierarchical attention module), 4) STAH model without temporal alignment attention.

4.3 Experiment Results

Method	MAPE		RMSE	
	FSL	DC	FSL	DC
Bottom-up MA	8.97	5.92	1.32	16.85
Bottom-up ARIMA	10.12	6.37	1.53	18.76
Baseline 3)	7.32	5.71	1.21	13.64
Baseline 4)	6.84	4.98	1.13	12.19
STAH	**6.63**	**4.79**	**1.04**	**10.86**

Table 1. The comparison of MAPE and RMSE obtained by different methods.

The prediction accuracy at each level and the average statistical results for the dataset are shown in Table 1, respectively. From the tables, we can see in general that the DC level prediction has less errors than the FSL level forecasts and our proposed model achieves the best performance in both levels. As we can see, Deep learning approaches generally achieved better prediction results than tradition models. Compared with baseline 3) that did not incorporate spatial topology, our model STAH has achieved a significant improvement. This demonstrates our model can effectively utilize spatial structure to make more accurate predictions.

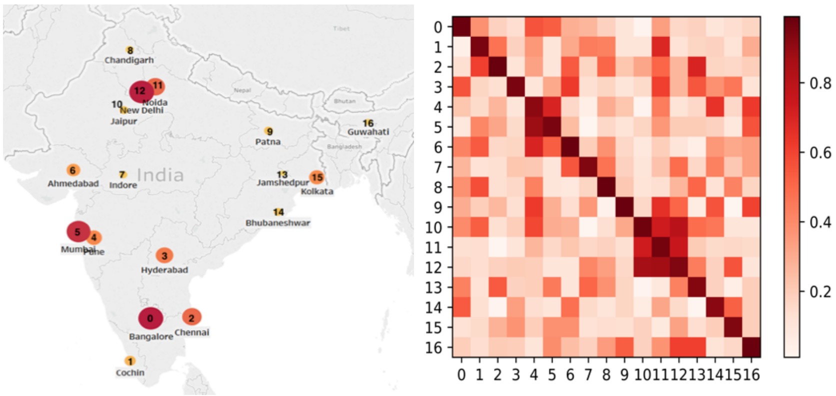

Figure 3. The spatial attention matrix obtained from spatial hierarchical attention module.

In order to investigate the role of spatial attention mechanism in our model intuitively, we perform a case study: picking out ten service parts belonging to the same category and showing the average spatial attention matrix of the DC graph structure in the training set. As shown on the right side of Figure 3, each element represents the correlation strength between the i-th FSL and the j-th FSL in India. The service part demand on the 10th FSL is closely related to the ones on the 11th and 12th FSLs. This is because they are close in space and customers in that regions who have service parts replacement requirements can go to one of them with no difference. This certainly explains why the spatial hierarchical attention module improves forecast accuracy and shows an interpretability advantage.

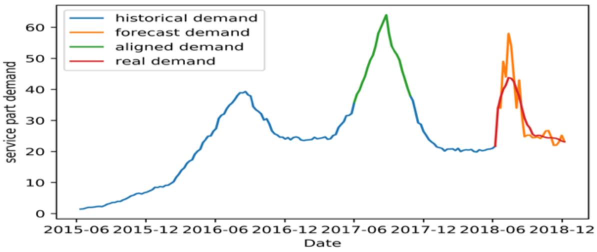

Figure 4. Demonstration of the temporal alignment attention mechanism in the model STAH by visualizing one service part demand at the DC level.

Compared with baseline 4), when removing the temporal alignment attention mechanism, the performance drops. So, we can conclude the temporal alignment attention contribute positively. Furthermore, we visualize one service part historical demand at the DC level together with the prediction results, as shown in Figure 4. It highlights the aligned segment of service part historical demand and predicted service part demand. We find that two highlighted parts have similar trend and seasonality.

m	MAPE		RMSE	
	FSL	DC	FSL	DC
0	6.82	4.98	1.17	11.14
2	6.76	4.92	1.13	11.06
5	6.69	4.83	1.08	10.92
10	**6.63**	**4.79**	**1.04**	**10.86**
100	6.73	4.87	1.12	11.36

Table 2. Demonstration the influence of the number of pairs of spatial attention matrixes in the REG term.

To study the effects of the hyperparameter m, we test 5 different values $m = 0, 2, 5, 10, 100$, as shown in Table 2. We observe the increase of the number of pairs of spatial attention matrixes m, the performance of the model first increases and then decreases. Larger m can decrease the flexibility and increase the generalization of the model at the cost of decrease model complexity and more prune to underfitting.

5 Conclusion and Future work

In this paper, we propose a novel deep learning framework STAH for hierarchical service part demand forecast. Experiments show that our model not only achieves better performances but also increase the interpretability. In the future, we will further apply this model on more complicated service supply chain networks that have more levels to test the model's potential capability. Moreover, our proposed model can be applied into more general spatial-temporal structured sequence forecasting scenarios, such as electricity demand, preference prediction in recommendation system, etc.

References

1. Bacchetti, A., Saccani, N.: Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. Omega. 40, 722–737 (2012).
2. Assaad, M., Boné, R., Cardot, H.: A new boosting algorithm for improved time-series forecasting with recurrent neural networks. Inf. Fusion. 9, 41–55 (2008).
3. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting & Control. (2015).
4. Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L.: Optimal combination forecasts for hierarchical time series. Comput. Stat. Data Anal. 55, 2579–2589 (2011).
5. Athanasopoulos, G., Ahmed, R.A., Hyndman, R.J.: Hierarchical forecasts for Australian domestic tourism. Int. J. Forecast. 25, 146–166 (2009).
6. de Livera, A.M., Hyndman, R.J., Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing. J. Am. Stat. Assoc. 106, 1513–1527 (2011).

7. Taieb, S. Ben, Rajagopal, R., Ben Taieb, S., Yu, J., Neves Barreto, M., Rajagopal, R.: Regularization in Hierarchical Time Series Forecasting With Application to Electricity Smart Meter Data. In: Conference on Artificial Intelligence (2017).
8. Wickramasuriya, S.L., Athanasopoulos, G., Hyndman, R.J.: Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization. J. Am. Stat. Assoc. 114, 804–819 (2018).
9. Niepert, M., Ahmed, M., Kutzkov, K.: Learning Convolution Neural Networks for Graphs. In: International conference on machine learning. pp. 2014–2023 (2016).
10. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral Networks and Locally Connected Networks on Graphs. In: International Conference on Learning Representations (2014).
11. Michaël Defferrard, Bresson, X., Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In: Advances in Neural Information Processing Systems. pp. 3844–3852 (2016).
12. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: International Conference on Learning Representations (2017).
13. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: International Conference on Machine Learning (2015).
14. Cheng, J., Dong, L., Lapata, M.: Long Short-Term Memory-Networks for Machine Reading. In: Conference on Empirical Methods on Natural Language Processing (2016).
15. Lin, Z., Feng, M., Santos, C.N. dos, Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A Structured Self-attentive Sentence Embedding. In: International Conference on Learning Representations (2017).
16. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: GRAPH ATTENTION NETWORKS. In: International Conference on Learning Representations (2018).
17. Liang, Y., Ke, S., Zhang, J., Yi, X., Zheng, Y.: Geoman: Multi-level attention networks for geo-sensory time series prediction. In: International Joint Conference on Artificial Intelligence. pp. 3428–3434 (2018).
18. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In: Conference on Artificial Intelligence (2019).
19. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. 30, 129–150 (2011).
20. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Comput. 9, 1735–1780 (1997).