Network Level Spatial Temporal Traffic State Forecasting with Hierarchical Attention LSTM (HierAttnLSTM)

Tianya Terry Zhang, Ph.D.
Research Assistant Professor
Center for Urban Informatics and Progress
University of Tennessee at Chattanooga
Chattanooga, TN, US, 37405
Email: tianya-zhang@utc.edu
ORCID: 0000-0002-7606-9886
Abstract

Traffic state data, such as speed, volume and travel time collected from ubiquitous traffic monitoring sensors require advanced network level analytics for forecasting and identifying significant traffic patterns. This paper leverages diverse traffic state datasets from the Caltrans Performance Measurement System (PeMS) hosted on the open benchmark and achieved promising performance compared to well recognized spatial-temporal models. Drawing inspiration from the success of hierarchical architectures in various Artificial Intelligence (AI) tasks, we integrate cell and hidden states from low-level to high-level Long Short-Term Memory (LSTM) networks with an attention pooling mechanism, similar to human perception systems. The developed hierarchical structure is designed to account for dependencies across different time scales, capturing the spatial-temporal correlations of network-level traffic states, enabling the prediction of traffic states for all corridors rather than a single link or route. The efficiency of designed attention-based LSTM is analyzed by ablation study. Comparative results with baseline LSTM models demonstrate that the Hierarchical Attention LSTM (HierAttnLSTM) model not only provides higher prediction accuracy but also effectively forecasts unusual congestion patterns. Data and code are made publicly available to support reproducible scientific research.

Main Text

1. Introduction

Traffic state information is a critical component of advanced traveler information systems (ATIS), which are extensively used for route guidance and mode choice. Short-term traffic state (less than 60-minute time horizon) prediction models are essential for trip planning, as they forecast traffic conditions in the near future to help users avoid unexpected delays. Continuous and updated traffic state data enables mobility management centers and commercial navigation apps to effectively adjust their forecasts of network congestion for travelers. From the users' perspective, predictive traffic state information is used to select routes, travel modes, and departure times based on perceived certainty. From the system perspective, predicting traffic states allows traffic system engineers to evaluate the potential benefits of various response strategies under different circumstances.

Traffic conditions are influenced by the imbalance between traffic demand and supply, traffic control measures, accidents, as well as external factors such as weather conditions and emergencies. Traditional time series models rely heavily on preprocessing and feature engineering, which is advantageous when the data volume is small. However, traditional statistical prediction methods, with their limited number of parameters, require frequent retraining and are thus inefficient for application across entire roadway networks. Recurrent neural networks (RNNs) address these limitations with automatic feature extraction capabilities. To mitigate the issues of gradient exploding/vanishing, gated mechanisms have been proposed in popular RNN architectures. The challenge of transportation spatial temporal prediction lies not only in complex temporal dependencies but also in capturing and modeling intricate, nonlocal, and nonlinear spatial dependencies.
between traffic conditions at various locations, which has not been fully addressed despite solutions for temporal dependencies like RNNs. Graph Neural Networks (GNNs) provide a novel approach to tackle transportation prediction challenges, especially in capturing and modeling complex spatial dependencies that traditional methods struggle with. Furthermore, their ability to learn dynamic graphs and incorporate diverse node/edge attributes provides richer feature representations, leading to more accurate and informed predictions.

Inspired by the success of state-of-the-art hierarchical spatial-temporal deep neural networks, we introduced a novel feature pooling layer to multi-layer Long Short-Term Memory (LSTM) networks that propagates cell and hidden states across multiple timesteps. Existing traffic prediction models using LSTM as the backbone typically employ a stacked architecture without cell-state hierarchical feature extraction capabilities. We proposed a hierarchical pooling module to capture information from different time steps, akin to the human perception system that consolidates low-level inputs into high-level abstractions, enhancing robustness and accuracy.

The motivation behind developing a hierarchical LSTM model for traffic state prediction stems from the observation that intelligent perceptual tasks, such as vision and language modeling, benefit from hierarchical representations. Features in successive stages become increasingly global, invariant, and abstract. This theoretical and empirical evidence suggests that a multi-stage hierarchy of representations can improve performance in understanding complex patterns and making accurate predictions. The hierarchical attention-pooling-based LSTM model is designed to learn representations at multiple levels of abstraction. Lower levels of the hierarchy capture local features and dependencies over short time intervals, while higher levels capture more global and long-term patterns. This design enables the model to understand complex temporal relationships, recurring traffic patterns, and other factors affecting traffic states.

2. Literature Review
   a. Traffic Forecasting Models
      The field of traffic forecasting has evolved significantly over the years, reflecting advancements in data analysis and computational techniques. Initially, traditional statistical methods were employed to predict traffic patterns, which were built on hand-engineered task-specific parameters include linear regression methods [1, 2], ARIMA [3,4], Kalman filter [5-7], Hidden Markov Models (HMMs) [8], and dynamic Bayesian networks [9]. As technology progressed, machine learning algorithms gained prominence, offering improved accuracy and the ability to handle more complex data. Machine Learning methods include Random Forests [10], support vector regression [11,12], k-Nearest Neighbor (KNN) Methods [13]. The congestion map-based method [14] combines historical data with real-time data to predict travel time. The historical data were classified with Gaussian Mixture Model and K-means algorithm to estimate congestion propagation using a consensual day. Dynamic linear models (DLMs) were designed [15] to approximate the non-linear traffic states. The DLMs assume their model parameters are constantly
changing over time, which is used to describe the Spatial-temporal characteristics of temporal traffic data.

The advent of deep learning marked a significant milestone, with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) demonstrating remarkable capabilities in capturing spatial and temporal dependencies in traffic data. Those data-driven approaches don't require location-specific info or strong modeling assumptions, which can fit into the constantly evolving temporal data analysis techniques. A Stacked Auto Encoder (SAE) [16] deep learning method for traffic flow prediction that leverages stacked autoencoders to learn generic traffic flow features, demonstrating superior performance compared to traditional methods. The CRS-ConvLSTM model [17] enhances short-term traffic prediction by identifying critical road sections through a spatiotemporal correlation algorithm and using their traffic speeds as input to a ConvLSTM network. The DMVST-Net [18] enhances taxi demand prediction by integrating temporal (LSTM), spatial (local CNN), and semantic views to capture complex non-linear spatial-temporal relationships in large-scale taxi demand data, outperforming existing methods that consider spatial and temporal aspects independently. The Sequence-to-sequence (Seq2Seq) RNN-based approaches can go beyond the univariate forecasting that outputs network scale travel time prediction. A stacked bidirectional LSTM [19] for network level traffic forecasting that handles missing values with imputation units. Given the success of the Attention mechanism in many fields, this study [20] integrated the attention mechanism with the LSTM model to construct the depth of LSTM and model the long-range dependence.

More recently, Spatial-Temporal Graph Neural Networks (GNNs) have emerged as powerful tools for modeling the inherent network structure of transportation systems, while attention-based architectures have shown promise in focusing on the most relevant features for prediction. DCRNN [21] models traffic flow as a diffusion process on a directed graph, capturing spatial dependencies through bidirectional random walks and temporal dependencies using an encoder-decoder architecture. The Graph WaveNet [22] architecture addresses limitations in spatial-temporal graph modeling by employing an adaptive dependency matrix to capture hidden spatial relationships and utilizing stacked dilated 1D convolutions to handle long temporal sequences. The ASTGCN [23] model enhances traffic flow forecasting by incorporating three independent components to capture recent, daily periodic, and weekly-periodic dependencies with spatial-temporal attention mechanisms. The GCGA [24] addresses the real-time traffic speed estimation problem with limited data, leveraging graph convolution and generative adversarial networks to effectively extract spatial features and generate accurate traffic speed maps. STSGCN model [25] improves spatial-temporal network data forecasting by simultaneously capturing complex localized spatial-temporal correlations and heterogeneities through a synchronous modeling mechanism and multiple time-period modules. The LSGCN [26] framework enhances both long-term and short-term traffic prediction by integrating a novel cosAtt graph attention network with graph convolution networks in a spatial gated block, combined with gated linear units convolution. The GMAN [27] enhances long-term
traffic prediction by utilizing an encoder-decoder architecture with multiple spatio-
temporal attention blocks and a transform attention layer. SimST \cite{28} replaces
computationally expensive Graph Neural Networks (GNNs) with efficient spatial context
injectors. This STPGNN \cite{29} introduces a pivotal node identification module, a pivotal
graph convolution module, and a parallel framework to effectively capture spatio-temporal
traffic features on both pivotal and non-pivotal nodes.

Researchers have also explored hybrid approaches (e.g., DNN-BTF \cite{30}, ST-GAT \cite{31},
Frigate \cite{32}), combining different methodologies to leverage their respective strengths and
address the multifaceted nature of traffic dynamics. Transformer-based models \cite{33-35},
Reinforcement Learning \cite{36}, ODE-based \cite{37, 38} and Generative Adversarial Networks
(GANs) \cite{39} were also applied to spatial-temporal traffic forecasting tasks. This ongoing
evolution reflects the continuous effort to improve the accuracy and reliability of traffic
state forecasting models, crucial for effective traffic management and urban planning.

b. Hierarchical Spatial-Temporal Modeling
Hierarchical deep learning architecture is a widely adopted framework for spatial-
temporal data analysis, which has been applied in many vision and language learning tasks
\cite{40}. Inspired by the success of pyramid feature extraction in computer vision, researchers
have tried similar approaches for time series data modeling, and many results have shown
great benefits by employing multiscale scheme for efficient video-summarization
applications. By incorporating temporal structure with deep ConvNets for video
representation for video content analysis, Hierarchical Recurrent Neural Encoder (HRNE)
\cite{41} is proposed that can efficiently exploit video temporal structure to model the temporal
transitions between frames as well as the transitions between segments. The Temporal
Shift Module (TSM) was proposed \cite{42} for hardware-efficient video streaming
understanding. TSM model has three main advantages: low latency inference, low
memory consumption, and multi-level fusion. A spatial-temporal action detection and
localization model \cite{43} using a Hierarchical LSTM and achieved the state-of-the-art in
spatial-temporal video analysis, which is a basic functional block for a holistic video
understanding and human-machine interaction. The multi-resolution convolutional
autoencoder (MrCAE) architecture \cite{44} models the Spatial-temporal dynamics using a
progressive-refinement strategy. A multiscale convolutional LSTM network
(MultiConvLSTM) \cite{45} was implemented for travel demand and Origin-Destination
predictions. Their experiments on real-world New York taxi data have shown that the
MultiConvLSTM considers both temporal and spatial correlations and outperforms the
existing methods. A deep hierarchical LSTM network \cite{46} for video summarization
(DHAVS) extracts spatial-temporal features and applied an attention-based hierarchical
LSTM module to capture the temporal dependencies among video frames. Hierarchical
spatial-temporal modeling was explored in smart manufacturing in characterizing and
monitoring global anomalies to improve higher product quality \cite{47}. The Hierarchical
Information Enhanced Spatio-Temporal (HIEST) \cite{48} prediction method improves traffic
forecasting by modeling sensor dependencies at regional and global levels, using Meta
Vision and language understanding task is deemed as the benchmark for evaluating progress in artificial intelligence. Given the impressive performance of hierarchical feature learning in various vision-language understanding applications, in the next section, we propose a novel hierarchical LSTM model for the short-term travel time prediction task. Compared to existing LSTM-based traffic state prediction models that only modify the data input layers for feature extraction; our new designed hierarchical LSTM model breaks the interconnections within the “black-box” neural networks with attention pooling strategy.

3. METHODS
   a. Preliminary
   In this section, we briefly describe the key components and variants of the building LSTM unit. LSTM is modified from the vanilla RNN (Recurrent Neural Network) model to enhance the capability of long-term temporal dependence for sequential feature extraction. LSTM has shown great performance on many language tasks or time-varying data modeling. The classic LSTM cell has led to several variants by adding new modifications, such as ConvLSTM [49], Grid LSTM [50], and Eidetic LSTM [51]. Three main gates were collectively used for progressively updating the output: Input Gate, Output Gate, and Forget Gate. The key feature of LSTM is the Cell State, which works as the memory pipe to transmit the long-term memory stored in the previous state to the current state. The input and forget gates are used as knobs to determine which information needs to be deleted or added to the cell state. Equation (5) describes how the current cell state adds or forgets information with the forget gate and the input gates. The output gate takes the inputs, newly updated long-term memory, and previous short-term memory to compute a new hidden state/short-term memory. The LSTM unit model (Figure 1) used in this paper is iterated as follows:

\[
\begin{align*}
    i_t &= \sigma(W_{ix}x_t + b_i + W_{ih}h_{t-1} + b_h) \\
    f_t &= \sigma(W_{fx}x_t + b_f + W_{fh}h_{t-1} + b_f) \\
    o_t &= \sigma(W_{ox}x_t + b_o + W_{oh}h_{t-1} + b_o) \\
    g_t &= \emptyset(W_{gx}x_t + b_g + W_{gh}h_{t-1} + b_g) \\
    c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
    h_t &= o_t \odot \emptyset(c_t)
\end{align*}
\]

where \( f_t \) is forget gate at timestamp \( t \), \( c_t \) is the cell state at timestamp \( t \), and \( o_t \) is the output gate at timestamp \( t \). \( \sigma \) represents a sigmoid operation and \( \emptyset \) represents \( \tanh \) activation function. \( \odot \) is the Hadamard product. \( W \) is the weight matrix that conducts affine transformation on the input \( x_t \) and hidden state \( h_t \). Matrices are depicted with capital letters while vectors with non-capital bold letters.
b. Cell and Hidden States Attention Pooling

In the proposed approach, the hierarchical LSTMs are devised at the higher layer by taking cell states and hidden states from the lower layer through an attention pooling module, which is shown in Figure 2. The attention pooling module computes the spatial-temporal representation of network travel time with a larger time duration which is analogous to the pooling operation in CNN. The devised module increases the temporal receptive field by carrying forward a compact representation of batch sequential inputs.

In the diagram, $S$ and $PS$ are shorthand for States and Pooled States, including Cell State ($CS$) and Hidden State ($HS$). The superscript $l$ and $l-1$ represent layer number. The subscript $i, i+1, ..., i+k$ represent sequential inputs. $n-1$ and $n$ are timesteps for
the top-layer LSTM. The dashed path is only used when we compute the cell state (CS). The governing equations for the interconnection of different LSTMs are modified as below.

With an affine transformation, each cell state (CS) and hidden state (HS) vector will be converted into a single number:

\[
u_{CS_{n-1}}^l = Affine(CS_{n-1}^l) \quad (7)
\]
\[
u_{HS_{i+t}^{l-1}} = Affine(HS_{i+t}^{l-1}) \quad (8)
\]
\[
u_{CS_{i+t}^{l-1}} = Affine(CS_{i+t}^{l-1}) \quad (9)
\]

The prefix \(u\) denotes a natural number. After that, the \(u_{CS_{n-1}}^l, u_{HS_{i+t}^{l-1}}, \) and \(u_{CS_{i+t}^{l-1}}\) will be sent to Softmax to compute weight factors.

\[
v_{CS_{n-1}}^l = \frac{\exp(u_{CS_{n-1}}^l)}{\sum_{t=1}^{K} \exp(u_{CS_{n-1}}^l)} \quad (10)
\]
\[
v_{CS_{i+t}^{l-1}} = \frac{\exp(v_{CS_{i+t}^{l-1}})}{\sum_{t=1}^{K} \exp(v_{CS_{i+t}^{l-1}})} \quad (11)
\]
\[
v_{HS_{i+t}^{l-1}} = \frac{\exp(u_{HS_{i+t}^{l-1}})}{\sum_{t=1}^{K} \exp(u_{HS_{i+t}^{l-1}})} \quad (12)
\]

The prefix \(v\) denotes the weight factor for corresponding Cell States and Hidden States after Softmax operation.

The Pooled Cell State (PCS) and Pooled Hidden State (PHS) are computed with the obtained weights times original cell state and hidden state vectors.

\[
PCS_n^l = v_{CS_{n-1}}^l * CS_{n-1}^l + \sum_{t=1}^{K} v_{CS_{i+t}^{l-1}} * CS_{i+t}^{l-1} \quad (13)
\]
\[
PHS_n^l = \sum_{t=1}^{K} v_{HS_{i+t}^{l-1}} * HS_{i+t}^{l-1} \quad (14)
\]

The top layer LSTM is updated with PCS and PHS by following equations:

\[
i_n^l = \sigma(W_{ix}^l PCS_n^l + b_{il}^l + W_{ih}^l h_{n-1}^l + b_{hi}^l) \quad (15)
\]
\[
f_n^l = \sigma(W_{fx}^l PHS_n^l + b_{if}^l + W_{fh}^l h_{n-1}^l + b_{hf}^l) \quad (16)
\]
\[
o_n^l = \sigma(W_{ox}^l PCS_n^l + b_{io}^l + W_{oh}^l h_{n-1}^l + b_{ho}^l) \quad (17)
\]
\[
g_n^l = \emptyset(W_{gx}^l PHS_n^l + b_{ig}^l + W_{gh}^l h_{n-1}^l + b_{hg}^l) \quad (18)
\]
\[
c_n^l = f_n^l \odot PCS_n^l + i_n^l \odot g_n^l \quad (19)
\]
\[
h_n^l = o_n^l \odot \emptyset(c_n^l) \quad (20)
\]

c. Hierarchical Attention LSTM (HierAttnLSTM) Structure

Before passing the features extracted from LSTM layers to the fully connected layer, the self-attention module is added that can further address the temporal dependence for time-varying data modeling. The entire structure of the proposed HierAttnLSTM model is
shown in Figure 3. The bottom layer processes the entire time sequence, while the upper layer computes the pooled hidden states with latent variables. The cell states that represent the longer memories from both bottom and top layer LSTMs are aggregated by the attention pooling to generate new cell states for top-layer LSTM. After the hierarchical pooled LSTMs, the hierarchically processed features were forwarded to a self-attention layer for dimension reduction. Then the attention layer output was subsequently sent to a fully connected layer to compute final prediction. The hidden states from lower-level LSTM are treated as new time sequences for top level LSTM, and the cell states of top-layer LSTM are a function of previous layer and its own cell states. In this way, multiscale spatial and temporal information is captured at different levels of LSTMs.

3. Experiment

a. Dataset

The dataset is collected from Caltrans Performance Measurement System (PeMS), an online system that continuously gathers real-world sensor data, offering a comprehensive and up-to-date representation of traffic conditions \[52\]. PeMS' public accessibility and widespread use in similar traffic systems allows for easy generalization of model results, increasing the practical impact to a broader range of real-world applications. We used the PeMS-Bay, PeMSD4, and PeMSD8 datasets standardized by the LibCity \[53\] benchmark. LibCity aims to provide researchers with a credible experimental tool and a convenient development framework, ensuring standardization and reproducibility in the field of traffic forecasting. The datasets PEMSD4, PEMSD8, and PEMS-BAY provide diverse traffic data for prediction tasks. PEMSD4 covers 307 nodes over 16,992 timesteps from January to February 2018, with flow, speed, and occupancy data. PEMSD8 includes 170 nodes over 17,856 timesteps from July to August 2016, also with flow, speed, and occupancy data. PEMS-BAY is the largest, with 325 nodes and 52,116 timesteps from January to May 2017, focusing solely on speed data. While these publicly available datasets cover
flow and speed, they lack travel time data, which was addressed by downloading an additional PEMS-Bay dataset for travel time prediction testing from January 2020 to October 2021.

b. Data Exploratory Analysis

Traffic sensor data provides a comprehensive view of variations in monitored area, highlighting clear patterns based on time of day, day of the week, and month of the year. These insights can inform traffic management strategies and help individuals plan their travel more efficiently. The PEMS-BAY area travel time data from 2020 reveals interesting patterns across different time scales. Monthly averages show relatively consistent median travel times throughout the year, with slightly more variability in the early months and lower times in the middle of the year. The monthly data suggests some seasonal effects, with winter months showing more variability. This could be due to weather conditions or holiday-related traffic patterns. Daily patterns demonstrate clear rush hour peaks on weekdays, with Friday evenings experiencing the highest travel times. Weekends exhibit a distinct pattern with later, less pronounced morning peaks and generally lower travel times compared to weekdays. (see Figure 4)

![Figure 4 Monthly and Daily Travel Time Pattern in PEMS District 4 data](image)

The main feature of time series is autocorrelation, which is the correlation for the data with itself at previous timestamps. It is the assumption of time series forecasting models and helps us reveal the underlying patterns, specifically, the seasonality and trends of travel time characteristics. The partial autocorrelation function (PACF) is similar to the ACF except that it displays only the correlation between two observations. Additionally, analyzing the ACF and PACF in conjunction is necessary for choosing the appropriate model for our time series prediction. A very high autocorrelation in travel time data has been identified after calculating autocorrelation and partial autocorrelation because traffic conditions 5 minutes ago will most likely affect the current travel time. As time increases, the correlation declines more and more (see Figure 5).
c. Implementation Details

A comprehensive approach is applied for hyperparameter tuning, systematically exploring different combinations of hidden sizes (64, 128, 256), number of layers (2, 3, 4), and attention hops (2, 3, 4), among other parameters. For each combination, the implementation creates a configuration dictionary with specific model parameters. The training process is managed through an executor configuration file, which specifies key training parameters. The designed model is trained on a Google Cloud A100 GPU, a maximum of 100 epochs with early stopping after 5 epochs of no improvement, and the use of the Adam optimizer with a MultiStepLR learning rate scheduler. The configuration also sets up logging, model saving, and evaluation criteria. This comprehensive setup, implemented using the LibCity library, enables a systematic exploration of the model's hyperparameter space, allowing researchers to identify the most effective configuration for their specific task while maintaining consistency in the training environment and methodology. After completing the hyperparameter tuning process, the best-performing model configuration was identified with a hidden size of 128, 3 layers, and 3 attention hops. This optimal configuration likely provides the best balance of model complexity and performance for the specific task at hand. Two performance metrics are used to evaluate the model’s performance. Mean Absolute Error (MAE) is used to measure model accuracy. Root Mean Square Error (RMSE) is sensitive to model stability.

\[ RMSE = \sqrt{\frac{1}{N+C} \sum_{j=1}^{C} \sum_{t=1}^{N} \left( \hat{T}_{t,j}(t) - T_{t,j}(t) \right)^2} \quad (21) \]

\[ MAE = \frac{1}{N+C} \sum_{j=1}^{C} \sum_{t=1}^{N} |\hat{T}_{t,j}(t) - T_{t,j}(t)| \quad (22) \]

where, \( T_{t,j}(t) \) and \( \hat{T}_{t,j}(t) \) are the predicted and ground truth travel time for corridor \( j \) at timestamp \( t \). \( C \) is the total number of corridors. \( N \) is the total number of timestamps in the output window.
4. Results

Table 1. PEMSD4 Traffic Flow Forecasting

| MODEL      | 3 STEP (15-min) | 6 STEP (30-min) | 9 STEP (45-min) | 12 STEP (60-min) |
|------------|-----------------|-----------------|-----------------|------------------|
|            | MAE             | RMSE            | MAE             | RMSE             |
| HierAttnLSTM | 9.079           | 22.766          | 8.933           | 22.574           |
| AGCRN [55]  | 18.132          | 29.221          | 18.834          | 30.464           |
| GWNET [22]  | 17.692          | 28.516          | 18.574          | 29.888           |
| MTGNN [54]  | 17.925          | 28.837          | 18.760          | 30.296           |
| GMAN [27]   | 18.790          | 29.549          | 19.538          | 30.805           |
| STGCN [56]  | 19.146          | 30.301          | 20.133          | 31.886           |
| GRU [57]    | 22.441          | 36.286          | 22.506          | 36.342           |
| Seq2Seq [58]| 22.585          | 36.475          | 22.581          | 36.348           |
| DCRNN [21]  | 19.581          | 31.125          | 21.467          | 34.067           |
| STG2Seq [60]| 23.006          | 35.973          | 23.251          | 36.227           |
| AE [59]     | 23.999          | 37.942          | 24.024          | 37.990           |
| ASTGCN [23] | 20.530          | 31.755          | 22.971          | 35.033           |
| TGCN [61]   | 21.678          | 34.635          | 23.962          | 37.777           |

Table 2. PEMSD8 Traffic Flow Forecasting

| MODEL      | 3 STEP (15-min) | 6 STEP (30-min) | 9 STEP (45-min) | 12 STEP (60-min) |
|------------|-----------------|-----------------|-----------------|------------------|
|            | MAE             | RMSE            | MAE             | RMSE             |
| HierAttnLSTM | 8.375           | 20.356          | 9.204           | 22.518           |
| GWNET [22]  | 13.486          | 21.615          | 14.349          | 23.375           |
| AGCRN [55]  | 14.146          | 22.241          | 14.962          | 24.055           |
| MTGNN [54]  | 14.001          | 21.988          | 14.883          | 23.624           |
| STGCN [56]  | 15.166          | 23.615          | 16.188          | 25.401           |
| GMAN [27]   | 15.158          | 23.021          | 15.924          | 24.553           |
| DCRNN [21]  | 15.139          | 23.476          | 16.619          | 25.982           |
| Seq2Seq [58]| 19.186          | 31.220          | 19.326          | 31.446           |
| GRU [57]    | 19.992          | 32.276          | 20.126          | 32.569           |
| STG2Seq [60]| 18.217          | 27.334          | 19.479          | 29.289           |
| AE [59]     | 22.266          | 35.562          | 22.209          | 35.557           |
| TGCN [61]   | 17.348          | 25.934          | 19.109          | 28.846           |

The performance of the HierAttnLSTM model on both PEMSD4 (Table 1) and PEMSD8 (Table 2) datasets demonstrates significant improvements over existing baseline models for traffic flow forecasting. Across all forecast horizons (15, 30, 45, and 60 minutes), HierAttnLSTM consistently outperforms the other 12 models in both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics. For PEMSD4, the model achieves MAE values ranging from 9.079 to 9.168 and RMSE values from 22.574 to 22.884 across different time steps, substantially outperforming the next best model, AGCRN.
Similarly, for PEMSD8, HierAttnLSTM shows remarkable performance with MAE values between 8.375 and 9.427, and RMSE values between 20.356 and 22.715. The improvements are particularly striking for shorter-term predictions, with the 3-step (15-minute) forecast showing nearly 50% reduction in MAE for PEMSD4 and about 38% for PEMSD8 compared to the next best models. These results indicate that HierAttnLSTM achieves state-of-the-art performance in traffic flow forecasting, offering substantial gains in prediction accuracy across different datasets and forecast horizons.

| MODEL      | 3 STEP (15-min) | 6 STEP (30-min) | 9 STEP (45-min) | 12 STEP (60-min) |
|------------|-----------------|-----------------|-----------------|------------------|
|            | MAE             | RMSE            | MAE             | RMSE             | MAE             | RMSE             | MAE             | RMSE             | MAE             | RMSE             |
| GWNET [22] | 1.317           | 2.782           | 1.635           | 3.704            | 1.802           | 4.154           | 1.914           | 4.404           |
| MTGNN [54] | 1.331           | 2.797           | 1.657           | 3.760            | 1.831           | 4.214           | 1.954           | 4.489           |
| DCRNN [21] | 1.314           | 2.775           | 1.652           | 3.777            | 1.841           | 4.301           | 1.966           | 4.600           |
| AGCRN [55] | 1.368           | 2.868           | 1.686           | 3.827            | 1.845           | 4.265           | 1.966           | 4.587           |
| STGCN [56] | 1.450           | 2.872           | 1.768           | 3.742            | 1.941           | 4.140           | 2.057           | 4.355           |
| GMAN [27]  | 1.521           | 2.950           | 1.828           | 3.733            | 1.998           | 4.107           | 2.115           | 4.321           |
| ASTGCN [23]| 1.497           | 3.024           | 1.954           | 4.091            | 2.253           | 4.708           | 2.522           | 5.172           |
| HierAttnLSTM | 2.493           | 5.163           | 2.496           | 5.177            | 2.779           | 5.494           | 2.587           | 5.340           |
| GRU [54]   | 2.491           | 5.204           | 2.508           | 5.288            | 2.535           | 5.384           | 2.575           | 5.510           |
| Seq2Seq [58]| 2.443           | 5.108           | 2.446           | 5.144            | 2.493           | 5.259           | 2.581           | 5.470           |
| AE [59]    | 2.570           | 5.302           | 2.573           | 5.288            | 2.627           | 5.392           | 2.724           | 5.608           |
| STG2Seq [60]| 2.192           | 4.231           | 2.424           | 4.826            | 2.604           | 5.266           | 2.768           | 5.650           |
| TGCN [61]  | 2.633           | 5.288           | 2.739           | 5.525            | 2.906           | 5.875           | 3.103           | 6.314           |

The results for PEMS-BAY (Table 3) traffic speed prediction reveal that graph-based models like GWNET, MTGNN, and DCRNN outperform the HierAttnLSTM model across all forecast horizons. This performance difference is likely due to PEMS-BAY's larger number of timesteps, which benefits from the spatial relationship modeling capabilities of graph-based approaches. While HierAttnLSTM shows consistent performance across time steps, it doesn't match the accuracy of graph-based models on this complex dataset. This outcome highlights a promising future research direction: combining graph models with the Hierarchical Attention LSTM approach. Such a hybrid model could potentially leverage the strengths of both architectures, addressing the current limitations on datasets with complex spatial relationships and improving performance on large-scale traffic networks like PEMS-BAY.

5. Ablation Study

In ablation study, we evaluated the effectiveness of our proposed HierAttnLSTM model against baseline deep learning models for travel time prediction. Additional data extracted from PEMS-BAY were downloaded with our own scraping tool, as public benchmarks lack travel time prediction datasets, we compared our designed LSTM model to vanilla LSTM models (unidirectional Stacked LSTM and Bidirectional Stacked LSTM) implemented without the attention pooling layer for Hidden and Cell States. The baseline models process spatial-temporal input data, with travel time information from all corridors.
fed at each time step. This comparison aims to isolate the impact of the Attention Pooling Layer in our HierAttnLSTM model, demonstrating its contribution to performance in travel time prediction tasks.

Given all $C$ corridors in the study area and 5-minute resolution data, the previous 2-hour travel time records of corridor $j$ are denoted as $\{t_{j, T-23}^T, ..., t_{j, T}^T\}$ ($j \in [1, C]$). The deep learning model output is the travel time at future time stamp $T + \delta t$ for all corridors $\{t_{1, T+\delta t}^T, t_{2, T+\delta t}^T, ..., t_{C, T+\delta t}^T\}$. The training, validation and testing dataset were randomly generated with sample sizes of 12000 (60%), 4000 (20%) and 4000 (20%). The comparison results are shown in Table 4.

Table 4. Ablation Analysis on Travel Time Prediction at Different Horizons

| Model        | 15 min MAE | 15 min RMSE | 30 min MAE | 30 min RMSE | 45 min MAE | 45 min RMSE |
|--------------|------------|-------------|------------|-------------|------------|-------------|
| Stacked LSTM | 0.247      | 0.445       | 0.272      | 0.517       | 0.286      | 0.557       |
| Stacked BiLSTM | 0.278      | 0.470       | 0.296      | 0.541       | 0.314      | 0.583       |
| HierAttnLSTM | 0.195      | 0.339       | 0.235      | 0.424       | 0.268      | 0.49        |

Our model has shown considerably better prediction results than existing LSTM-based travel time prediction results (Table 4). In Figure 6, sample travel time prediction results from different prediction horizons are presented for one-week period. Our proposed model demonstrates significant advantages over two other LSTM-based baselines, after removing the Hierarchical Attention Pooling. More specifically, the HierAttnLSTM can correctly predict the high spikes in travel time in the extended future, which is often the most desirable functionality of travel time prediction model. While the comparable models tend to underestimate the unexpected congestion and fall short of predicting the sudden spikes. The hierarchical attention pooling enhanced the spatial-temporal receptive fields of different levels of LSTM units, which augmented the model’s capacity for capturing unusual traffic patterns. The result indicates that adding hierarchical information with attention pooling to distill the cell states of LSTMs could successfully improve the travel time forecasting accuracy.
5. Conclusions

The ability to learn hierarchical representations automatically from the data makes the Hierarchical-Attention-LSTM traffic state prediction model a powerful tool for developing accurate and robust travel information prediction systems. From the model design perspective, this paper adds hierarchical feature pooling to the multi-layer LSTM and demonstrates superior prediction accuracy. The proposed Cell and Hidden states pooling architecture ensures that only important features are forwarded from lower to higher layers, mimicking the multiscale information abstraction of the human brain and adaptable to other spatial-temporal learning tasks. Unlike traditional LSTM-based models, our approach redesigns the internal structure of multi-layer LSTM by introducing attention pooling. This allows the model to better focus on relevant information through novel attention pooling modules. The attention mechanism selectively emphasizes or downplays hidden and cell states based on their importance for predictions, comprehensively leveraging information stored in LSTM cells and improving retention of important contextual information over time.

Importantly, we tested our model on different traffic state prediction tasks: traffic flow, speed, and travel time, using both publicly accessible datasets and our own scraped dataset. This comprehensive evaluation demonstrates the model's versatility and effectiveness across various traffic prediction challenges. Furthermore, we conducted a thorough analysis through ablation studies, clearly demonstrating the effectiveness of
adding Attention Pooling to multilayer LSTM. This validation reinforces the key innovation of our approach and its contribution to improved performance. Testing results show the proposed model exhibits the capability to predict unusual spikes in travel time caused by traffic congestion. This crucial finding indicates better generalization to unseen data and more reliable predictions in real-world scenarios.

For future research, exploring additional roadway information with Graph Neural Network models could further enhance the translation of multi-source data inputs into more abstract representations, potentially leading to even more accurate and robust traffic prediction systems.

7. Author contributions
The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

8. Acknowledgments
This research is supported by The Federal Highway Administration (FHWA) Exploratory Advanced Research (EAR) Program. Award No.: 693JJ32350028. The authors wish to thank Prof. Zack Ives at the University of Pennsylvania for his teaching Big Data Analytics. Thanks to Ying Ye as a team member of the project.

9. Conflict of interest
The authors declare that they have no conflict of interest.

10. Data availability
The data that support the findings of this study are available in the GitHub repository. These data were derived from the following resources available in the public domain: https://github.com/TeRyZh/Network-Level-Travel-Prediction-Hierarchical-Attention-LSTM.

11. References
1. Vlahogianni EI, Karlaftis MG, Golias JC. 2014. Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies 43, 3-19.
2. Zhang X, Rice J. 2003. Short-term travel time prediction. Transportation Research Part C: Emerging Technologies 11(3-4), 187-210.
3. Williams BM, Hoel LA. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of Transportation Engineering 129(6), 664-72.
4. Smith BL, Williams BM, Oswald RK. 2002. Comparison of parametric and nonparametric models for traffic flow forecasting. Transportation Research Part C: Emerging Technologies 10(4), 303-21.
5. Van Lint JW. 2008. Online learning solutions for freeway travel time prediction. IEEE Transactions on Intelligent Transportation Systems 9(1), 38-47.
6. Liu X, Chien SI, Chen M. 2015. An adaptive model for highway travel time prediction. Journal of Advanced Transportation 48, 642-654.
7. de Bézenac E, Rangaparam SS, Benidis K, Bohlke-Schneider M, Kurle R, Stella L, Hasson H, Gallinari P, Januschowski T. 2020. Normalizing Kalman Filters for Multivariate Time Series Analysis. NeurIPS 33.
8. Cheng YC, Li ST. 2011. Fuzzy time series forecasting with a probabilistic smoothing hidden Markov model. IEEE Transactions on Fuzzy Systems 20(2), 291-304.
9. Robinson JW, Hartemink AJ, Ghahramani Z. 2010. Learning Non-Stationary Dynamic Bayesian Networks. Journal of Machine Learning Research 11(12).
10. Zhang Y, Haghani A. 2015. A gradient boosting method to improve travel time prediction. Transportation Research Part C 58, 308-324.
11. Castro-Neto M, Jeong YS, Jeong MK, Han LD. 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications 36(3), 6164-6173.
12. Gao P, Hu J, Zhou H, Zhang Y. 2016. Travel time prediction with immune genetic algorithm and support vector regression. Proceedings of the 12th World Congress on Intelligent Control and Automation (WCICA), 987-992.
13. Zhao J, Gao Y, Tang J, Zhu L, Ma J. 2018. Highway travel time prediction using sparse tensor completion tactics and nearest neighbor pattern matching method. Journal of Advanced Transportation 2018.
14. Chiabaut N, Faitout R. 2021. Traffic congestion and travel time prediction based on historical congestion maps and identification of consensual days. Transportation Research Part C: Emerging Technologies 124, 102920.
15. Kwak S, Geroliminis N. 2020. Travel time prediction for congested freeways with a dynamic linear model. IEEE Transactions on Intelligent Transportation Systems.
16. Lv Y, Duan Y, Kang W, Li Z, Wang FY. 2014. Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems 16(2), 865-873.
17. Yang G, Wang Y, Yu H, Ren Y, Xie J. 2018. Short-term traffic state prediction based on the spatiotemporal features of critical road sections. Sensors 18(7), 2287.
18. Yao H et al. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. Proceedings of the AAAI Conference on Artificial Intelligence 32(1), 1-25.
19. Cui Z, Ke R, Pu Z, Wang Y. 2020. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. Transportation Research Part C: Emerging Technologies 118, 102674.
20. Ran X, Shan Z, Fang Y, Lin C. 2019. An LSTM-based method with attention mechanism for travel time prediction. Sensors 19(4), 861.
21. Li Y, Yu R, Shahabi C, Liu Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. ICLR.
22. Wu Z, Pan S, Long G, Jiang J, Zhang C. 2019. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121.
23. Guo S, Lin Y, Feng N, Song C, Wan H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 33(01), 922-929.
24. Yu JJQ, Gu J. 2019. Real-time traffic speed estimation with graph convolutional generative autoencoder. IEEE Transactions on Intelligent Transportation Systems 20(10), 3940-3951.
25. Song C, Lin Y, Guo S, Wan H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 34(01), 914-921.
26. Huang R, Huang C, Liu Y, Dai G, Kong W. 2020. LSGCN: Long short-term traffic prediction with graph convolutional networks. IJCAI 7, 2355-2361.
27. Zheng C, Fan X, Wang C, Qi J. 2020. Gman: A graph multi-attention network for traffic prediction. Proceedings of the AAAI Conference on Artificial Intelligence 34(01), 1234-1241.
28. Liu, X., Liang, Y., Huang, C., Hu, H., Cao, Y., Hooi, B. and Zimmermann, R., 2023. Do we really need graph neural networks for traffic forecasting?. arXiv preprint arXiv:2301.12603.
29. Kong W, Guo Z, Liu Y. 2024. Spatio-Temporal Pivotal Graph Neural Networks for Traffic Flow Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 38(08), 8627-8635.
30. Wu Y, Tan H, Qin L, Ran B, Jiang Z. 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transportation Research Part C: Emerging Technologies 90, 166-180.
31. Song J, Son J, Seo DH, Han K, Kim N, Kim SW. 2022. ST-GAT: A Spatio-Temporal Graph Attention Network for Accurate Traffic Speed Prediction. CIKM.
32. Gupta M, Kodamana H, Ranu S. 2023. Frigate: Frugal spatio-temporal forecasting on road networks. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 649-660.
33. Yan H, Ma X, Pu Z. 2022. Learning Dynamic and Hierarchical Traffic Spatiotemporal Features With Transformer. IEEE Transactions on Intelligent Transportation Systems 23(11), 22386-22399.
34. Cao H, Huang Z, Yao T, Wang J, He H, Wang Y. 2023. Inparformer: Evolutionary decomposition transformers with interactive parallel attention for long-term time series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 37(6), 6906-6915.
35. Li Z, Xia L, Xu Y, Huang C. 2024. GPT-ST: generative pre-training of spatiotemporal graph neural networks. Advances in Neural Information Processing Systems 36.
36. Liu F, Zhang W, Liu H. 2023. Robust Spatiotemporal Traffic Forecasting with Reinforced Dynamic Adversarial Training. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 1417-1428.
37. Yao C, Li Z, Wang J. 2023. Spatio-Temporal Hypergraph Neural ODE Network for Traffic Forecasting. IEEE International Conference on Data Mining (ICDM), 1499-1504.
38. Fang Z, Long Q, Song G, Xie K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 364-373.

39. Jiang J, Wu B, Chen L, Zhang K, Kim S. 2023. Enhancing the Robustness via Adversarial Learning and Joint Spatial-Temporal Embeddings in Traffic Forecasting. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM ’23).

40. Chung J, Ahn S, Bengio Y. 2016. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704.

41. Pan P, Xu Z, Yang Y, Wu F, Zhuang Y. 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1029-1038.

42. Lin J, Gan C, Han S. 2019. Tsm: Temporal shift module for efficient video understanding. Proceedings of the IEEE/CVF International Conference on Computer Vision, 7083-7093.

43. Ramaswamy A, Seemakurthy K, Gubbi J, Purushothaman B. 2020. Spatio-temporal action detection and localization using a hierarchical LSTM. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 764-765.

44. Liu, Y., Ponce, C., Brunton, S.L. and Kutz, J.N., 2023. Multiresolution convolutional autoencoders. Journal of Computational Physics, 474, p.111801.

45. Chu KF, Lam AY, Li VO. 2018. Travel demand prediction using deep multi-scale convolutional LSTM network. 21st International Conference on Intelligent Transportation Systems (ITSC), 1402-1407.

46. Lin J, Zhong SH, Fares A. 2022. Deep hierarchical LSTM networks with attention for video summarization. Computers & Electrical Engineering 97, 107618.

47. Guo S, Guo WG, Bain L. 2020. Hierarchical spatial-temporal modeling and monitoring of melt pool evolution in laser-based additive manufacturing. IIESE Transactions 52(9), 977-97.

48. Ma Q, Zhang Z, Zhao X, Li H, Zhao H, Wang Y, Liu Z, Wang W. 2023. Rethinking sensors modeling: Hierarchical information enhanced traffic forecasting. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 1756-1765.

49. Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo W. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems, 802-810.

50. Kalchbrenner N, Danihelka I, Graves A. 2015. Grid long short-term memory. arXiv preprint arXiv:1507.01526.

51. Wang Y, Jiang L, Yang MH, Li LJ, Long M, Fei-Fei L. 2019. Eidetic 3d lstm: A model for video prediction and beyond.

52. Liu X, Xia Y, Liang Y, Hu J, Wang Y, Bai L, Huang C, Liu Z, Hooi B, Zimmermann R. 2024. Largest: A benchmark dataset for large-scale traffic forecasting. Advances in Neural Information Processing Systems 36.
53. Wang J, Jiang J, Jiang W, Li C, Zhao WX. 2021. Libcity: An open library for traffic prediction. Proceedings of the 29th International Conference on Advances in Geographic Information Systems, 145-148.
54. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 753-763.
55. Bai L, Yao L, Li C, Wang X, Wang C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in Neural Information Processing Systems 33, 17804-17815.
56. Yu B, Yin H, Zhu Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.
57. Fu R, Zhang Z, Li L. 2016. Using LSTM and GRU neural network methods for traffic flow prediction. 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 324-328.
58. Sutskever I, Vinyals O, Le QV. 2014. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems 27.
59. Wang W, Huang Y, Wang Y, Wang L. 2014. Generalized autoencoder: A neural network framework for dimensionality reduction. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 490-497.
60. Bai L, Yao L, Kanhere S, Wang X, Sheng Q. 2019. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. arXiv preprint arXiv:1905.10069.
61. 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 21(9), 3848-3858.