Ensembling Spatial-Temporal Traffic Data

Nicholas Pang
School of Engineering, University of Connecticut
Storrs, Connecticut
nicholas.pang@uconn.edu

Daniel McKinney
School of Engineering, University of Connecticut
Storrs, Connecticut
daniel.mckinney@uconn.edu

I. Introduction

With the ever-increasing amount of vehicles on roads, traffic congestion has become a major problem in many cities around the world. Traffic congestion not only wastes time and fuel, but also has a negative impact on the environment and the economy. Predicting live traffic flow accurately is crucial for effective traffic management, especially during peak hours. It allows transportation departments to make more informed decisions, such as adjusting signal timings, managing traffic flow, and diverting traffic to alternative routes. Traffic congestion can then be minimized and drivers can experience smoother, safer, and more predictable driving conditions. Additionally, the benefits of accurate traffic prediction extend beyond individual drivers to businesses, transportation companies, and logistics providers. Predicting traffic flow can lead these organizations to optimize their operations, reduce delivery times, and improve efficiency.

Using a model for traffic forecasting is an example of a spatial-temporal prediction problem. Traffic is affected through spatial correlations due to the proximity of locations affecting each other, and through temporal correlations as a result of the changing traffic flows through time. More traditional methods of predicting traffic flow, such as statistical models or regression analysis, are limited in their accuracy as they fail to capture these complex spatial-temporal dependencies. More recent advancements in deep learning techniques have shown great potential in predicting live traffic flow accurately and efficiently

[1] proposes one such model, ASTGCN. ASTGCN uses a spatiotemporal graph structure to model the relationships between different locations and their associated traffic flow over time. The model applies graph convolutional layers to the graph structure to capture different spatial dependencies, while the recurrent neural network layer captures temporal dependencies in the data. In addition to these basic layers, ASTGCN includes attention mechanisms that allow it to selectively focus on important spatial and temporal features in the data.

ASTGCN, as well as other models based around the PeMS datasets, are modeled to predict traffic based on the spatial-temporal correlations within highway traffic. However, traffic is a complex problem that can be affected by factors such as road construction, weather conditions, pedestrian density, and accidents, all of which are not explicitly considered in the current model. In this paper, we will be designing an implementation to include pedestrian density and weather, two external factors which have direct impacts on traffic flow. Incorporating such external factors into the model can further enhance its predictive capabilities and make it more practical for real-world traffic management applications.

In order to efficiently incorporate the different datasets representing pedestrian count and weather updates, an ensemble approach is likely the ideal approach. Ensemble implementations are a powerful way to combine multiple datasets and improve the accuracy of a machine-learning model. The various proposed approaches and implementations of the ensemble will be further explored in the related work section. In the case of ASTGCN, adding more relevant datasets can provide additional information that can help the model make better overall predictions. With this specific approach, there are inevitable future challenges and hardships that may arise. One of which will likely happen sooner rather than later is data compatibility. The current model only trains live traffic data, meaning the model was designed and built for specific datasets. Tailoring the model to accommodate and accept the new datasets as trainable will likely require extra work.

Another preemptive hardship includes the method of ensemble selection. There exist different methods for combining the outputs of the individual models in an ensemble, and selecting the right method can be a challenge. Some methods, such as simple averaging or stacking, may be more straightforward to implement but may not be as effective as more complex methods like gradient boosting. Finally, the potential setback due to hardware is a plausible scenario as well. Ensembles, often computationally intensive, require significant processing power and memory to train and run.

The current plan of study can be broken into two parts: ensemble implementation and data modification. Though not inclusive of all preemptive threats to progress, the plan of study still includes the two major areas of research and design. For research on various ensemble implementations, it will largely align with information taken from the papers outlined in the paper survey. This area of research will involve studying similar models with ensemble implementations through literature review, brainstorming different applications, and ultimately testing the proposed datasets for each direction. Similarly, the other area of research on datasets will more so be focusing on internet exploration and extrapolation of viable data.

II. Related Work

A. ST-MetaNet

Our first proposed paper, ST-MetaNet [2], aims to tackle the same problem of traffic forecasting through means alike to ASTGCN. One of the two prime takeaways amongst the two models includes their graph-based representations. In ASTGCN, nodes are used to represent the spatial temporal locations while ST-MetaNet uses graph nodes to visualize the spatial relationships between different variables. The second takeaway that can be deduced from the paper is the end-goal approach. ST-MetaNet utilizes a meta-learning approach that, although less accurate, can be applied to other applications broader than traffic flow. ASTGCN, though producing more accurate predictions, is specifically designed to only be used for traffic forecasting.

Unfortunately, using ST-MetaNet as the baseline model was invariably impossible. The native OS used for the model is Linux, using an old version of Python, and with neither team member owning a Linux machine, and Linux VMs not being able to make up for that, we ultimately had to make the decision to drop ST-MetaNet and look for a paper with code that was executable on our native machines. This posed a major challenge, as the Linux VM was promising enough to continue pursuing ST-MetaNet, so by the time the decision was made to change models, there were only two days left.

B. Model Structure

DCRNN [5], a solid model for prediction problems, is a simple, structured network that effectively combines three common techniques for modeling: Diffusion, Convolution, and Recurrence. Between the models, both AST-GCN and DCRNN use the same convolutional operation for spatial aggregation, which is a graph convolution operation. They also take into account the sparsity of the graph structure when performing these graph convolutions to improve their computational efficiency. DCRNN provided useful due to its effective combination of three common approaches and analyzing their implementation, as well as further analysis of the graph structure.

As for recurrence within ASTGCN, the model largely alludes to techniques outlined in [3]. The RNN component of ASTGCN is adapted to the spatiotemporal graph structure of the traffic data by using graph convolutions to capture the spatial dependencies and recurrent connections to capture the temporal dependencies.

Another predecessor of ASTGCN includes STGCN. The STGCN component of ASTGCN is inspired by previous work on STGCNs for video action recognition [4]. The STGCN component of ASTGCN is adapted to the traffic flow prediction problem by incorporating additional graph attention mechanisms and gating mechanisms to capture the spatial and temporal dependencies in the traffic data.

The key advantage of ASTGCN is its new proposed direction is its ability to incorporate new datasets, such as weather and possible other factors, into its predictions. By doing so, the model can account for external factors that can impact traffic flow, such as rain, snow, or road closures due to construction. Thus making ASTGCN more robust and practical for real-world traffic management applications.

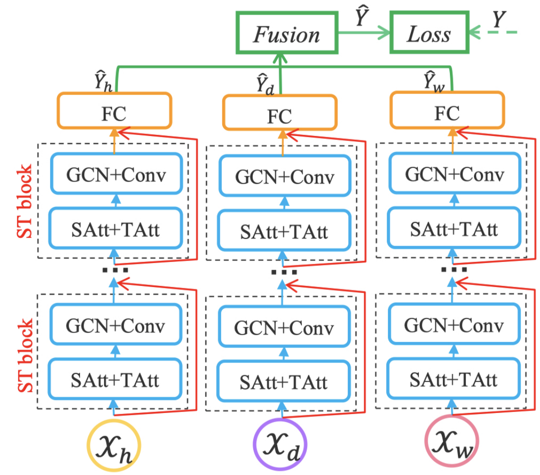

Fig. 1. ASTGCN Technical Markup

Incorporating these new datasets into the model requires an ensemble approach, where multiple datasets are combined to improve the accuracy of the machine learning model. This approach is particularly useful for ASTGCN since it can provide additional information that can help the model make better overall predictions, and is not a common trait amongst traffic forecasting models.

Compared to other traffic forecasting models, ASTGCN is unique in its ability to effectively capture the spatiotemporal dependencies in traffic data. By using graph convolutional layers and recurrent neural network layers, the model can effectively capture both the spatial and temporal aspects of the data. This is particularly useful for incorporating new datasets. Moreover, the attention and gating mechanisms in the model, also a unique feature, further enhance its predictive capabilities, allowing it to selectively focus on important spatial and temporal features in the data.

Overall, the ability of ASTGCN to incorporate new datasets and capture spatiotemporal dependencies in traffic data sets it apart from other traffic forecasting models. It has the potential to significantly improve traffic flow management and reduce congestion with the right data extrapolation.

C. Ensemble

Ensembles describe the combination of several models side by side in order to compute their output. Training ensemble models can be represented with two extremes. Independent ensemble training is when each model is trained separately, and the fully trained models are used to train the fusion method. End-to-end training is the process of training all models in tandem, where the loss is calculated after the fusion and back-propagated into the rest of the model. ASTGCN uses

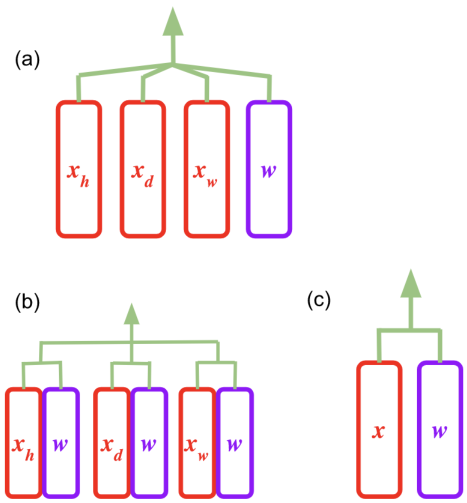

Fig. 2. Proposed Ensemble Technical Markup

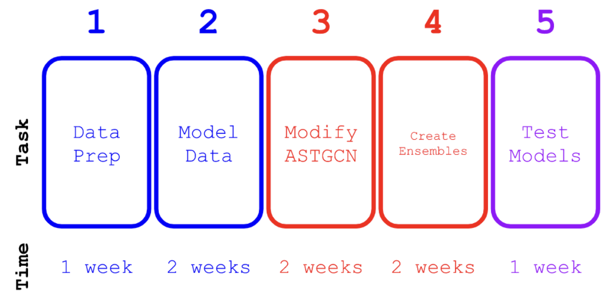

Fig. 3. Proposed Ensemble Technical Markup

end-to-end ensemble training for its three component modules, as seen in Fig. 1.

Recently, however, there have been debates on the effectiveness of end-to-end training within deep learning. [6] is an analysis of the choice between end-to-end and independent training concerning deep learning models. The conclusion finds that there is a strong correlation between poor end-to-end trained models and models that are over-parameterized.

III. Proposed Direction

Fig. 1 provides a description of ASTGCN. Note the three differing modules with x_h, x_d, and x_w as inputs. These represent spatiotemporal dependencies over three different timeframes - hourly, daily, and weekly. Each module produces its own prediction, which, in turn, is passed into a simple data fusion module and trained end-to-end. The module components have been simplified in Fig. 2 to better understand the proposed modifications to the original model.

Fig. 2(a) describes the highway traffic components and the weather components being fused together. Fig. 2(b) describes each highway traffic component being fused individually with the weather component, and then fusing them all together. Fig. 2(c) describes fusing all highway traffic components first before fusing them with the weather component. These diagrams do not encompass all possible configurations, nor do they account for any additional variables that may be considered. Additionally, each diagram will be tested with end-to-end training, independent training, and other additional ensemble training methods between the two.

Predicting traffic weather data is distinctly different from predicting weather, so using a prediction-based algorithm is not necessarily the best route to take. Instead, it is more intuitive to treat the weather as a known fact and use a weather forecast to classify the traffic conditions at a given time. One such example of a model that incorporates weather data into its algorithm to help produce accurate traffic condition outputs is the Deep Belief Network [7].

IV. Datasets

The proposed model will utilize the same highway traffic datasets as ASTGCN, namely the data provided by the Caltrans Performance Measurement System (PeMS). PeMS collects data through over 39,000 detectors spanned across highways throughout California. Measurements are taken every 30 seconds and take in a variety of factors. Two datasets derived from this data, PeMSD4 and PeMSD8, will be used in order to validate the model. PeMSD4 contains data from the San Francisco Bay area and consists of 3848 detectors through the period of January to February 2018. PeMSD8 contains data from San Bernardino and consists of 1979 detectors through the period of July to August 2016. Both datasets aggregate the raw PeMS data into 5-minute intervals and measure the total flow, average speed, and average occupancy.

At the present, we are unaware of any pre-prepared weather datasets that cover the above locations for those specific periods. The proposed solution is to retrieve weather data from Weather Underground, a site that allows data usage for non-commercial purposes. Weather Underground provides historical data measured by sensors throughout the world, with time periods varying depending on the source. To maintain consistent locations and time periods with PeMS, the following two stations will be used. The San Francisco International Airport (KSFO) provides weather updates every hour and The San Bernardino International Airport (KSBD) provides weather updates every 20 minutes. Both weather stations record data such as temperature, humidity, wind speed, precipitation, and sky condition. As this is inconsistent with the 5-minute periods in the PeMS datasets, the weather data will be reused under the root assumption that weather doesn't change drastically within those timeframes.

Additional datasets would be beneficial to test but will require a change in traffic datasets due to a lack of data within the San Francisco and San Bernardino areas with a possible candidate being New York City.

V. Timeline

In order to properly pace this project, it has been broken down into five steps, excluding presentation preparation. At least a week's worth of time is allocated to either end of the timeline for the proposal presentation and the final presentation respectively. Midterm reports and other writing will be done throughout this schedule as required.

The first step is to prepare relevant data. Data from weather underground must be scraped and cleaned in order to include it in the proposed model. Additionally, we are actively looking for other sources of data that may be relevant to traffic forecasting, but have not obtained any at the present.

The second step is to find simple models to evaluate the newly introduced variables with respect to traffic forecasting. Although these models will be tested for accuracy, the importance of said accuracy is not as important as the overall accuracy of the ensemble, and thus there is little emphasis in pursuing a highly accurate model.

The third step is to understand the source code of ASTGCN and determine methods to implement the proposed changes. The goal is to build a framework to allow for the addition of different joining methods to be easily implemented for the following step.

Once this framework is built, the fourth step is to build the variations in ensemble methods that will be evaluated in this project. The idea is to explore the subject area of ensembles further and discover new ways to perform data fusion across the models. By the end of this step, several different ensemble methods should be implemented and ready to test.

The final step of the project is to evaluate the ensembles. The baseline for evaluation will be the original ASTGCN itself, as the purpose of exploring ensemble methods is to improve the training of this specific model rather than an improvement to the core components. The models will be tested on root mean squared error (RMSE) and mean absolute error (MAE). Additional metrics may be implemented as further research into ensemble evaluation methods are explored.

VI. Responsibilities

Both students collaborated together to write and edit the vast majority of the proposal paper. In terms of sections of the paper, the introduction and half of the related work was written primarily by Daniel McKinney, and the other half of related work, proposed directions, datasets, and timeline was written primarily by Nicholas Pang.

References

[1] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting", AAAI, vol. 33, no. 01, pp. 922-929, Jul. 2019.

[2] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, 'Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning', in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 2019, pp. 1720–1730.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, 'Sequence to Sequence Learning with Neural Networks', CoRR, vol. abs/1409.3215, 2014.

[4] S. Chen, K. Xu, X. Jiang, and T. Sun, 'Spatiotemporal-Spectral Graph Convolutional Networks For Skeleton-Based Action Recognition', in 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2021, pp. 1–6.

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, 'Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting', CoRR, vol. abs/1707.01926, 2017.

[6] A. Webb et al., 'To ensemble or not ensemble: When does end-to-end training fail?', in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III, 2021, pp. 109–123.

[7] A. Koesdwiady, R. Soua, and F. Karray, 'Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach', IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9508–9517, 2016.