Traffic forecasting with spatiotemporal convolutional neural networks

Colton Cunov, Andreas Spetz, Oskar Thune, and Mathias Örtenberg Toftås
Chalmers University of Technology

(Dated: June 7, 2021)

Short term traffic forecasting is important to both the industry and citizens. It has been handled by statistical models until recently, when Neural Network approaches has increased in popularity. In this paper we apply a Spatio-Temporal Graph Neural Network to traffic data from the I-35 between Dallas and San Antonio. The goal of the project is making short term predictions of traffic volume along the highway from lightly processed data.

I. INTRODUCTION

Whether it's people trying to get to work on time or city planners, traffic is an issue that interests many parties. Avoiding delays caused by congestion is an important issue that affects citizens as well as organizations, and with increasing urbanization it becomes more critical as time goes on. The interest in developing these models is mainly to improve Intelligent Transportation Systems, but systems that help improve consumer quality-of-life are of interest to the public and thus a further reason for researching this topic.

Thus predicting congestion is a topic that has attracted quite a lot of attention since research started in the 1980:s [1]. Early on focus was laid on simple statistical models but with the rising interest in machine learning many neural network approaches to traffic prediction have been tried [2] in the last decade. In this paper we approach this subject by using a Spatio-Temporal Graph Neural Network [3], and comparing with the industry standard model ARMA [4]. The goal is for our model to outperform the ARMA model in terms of predicting the traffic speed along the given route.

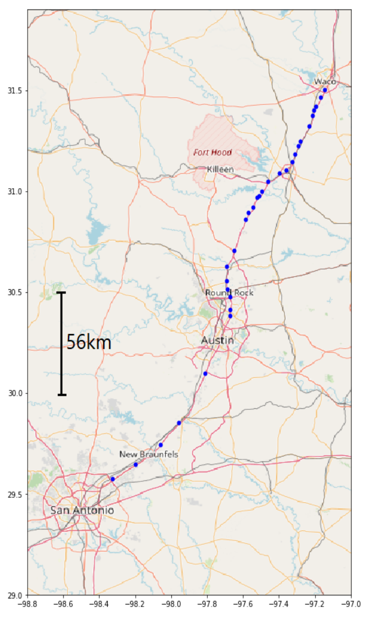

Figure 1. Sensor locations in the scope-limited dataset, with poor-performing (see Fig (3)) sensors omitted.

II. DATASET

Along a 350km stretch of Interstate 35 between Dallas and San Antonio, Texas, the Texas A&M Transportation Institute maintains infrastructure that reads Bluetooth signatures of passing cars. These readings are then used to compile data aggregations regarding traffic volume and average speeds along the highway, similar to the PeMS dataset.

We were provided 5-minute aggregations for 2019 and to limit scope, we chose to use the northbound direction and omit the sensors north of the IH-35 bifurcation (into IH-35E and IH-35W) south of the Dallas-Fort Worth metroplex. Although the speed readings are calculated from the distance between and arrival time at two sensors, our model requires node (not edge) values so we assign the speed to the destination node.

To examine temporal correlation, fig. (2) shows the average autocorrelation over all nodes for lags up to one week, with distinct seasonality of period 1 day. Considering this, we investigated the spatial correlation of nodes without respect for time, fig. (3). The tendency for immediately neighboring nodes to be correlated was visible for time lags up to four hours, but this likely holds for much longer.

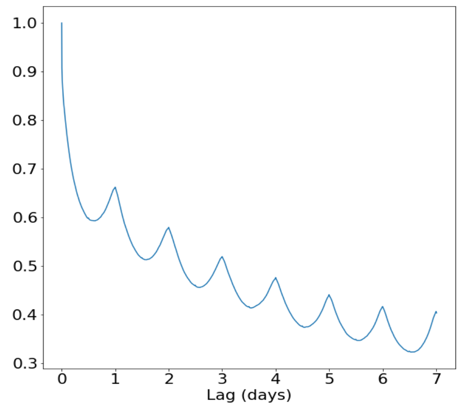

Figure 2. The mean autocorrelation over well-performing sensors for lags up to one week.

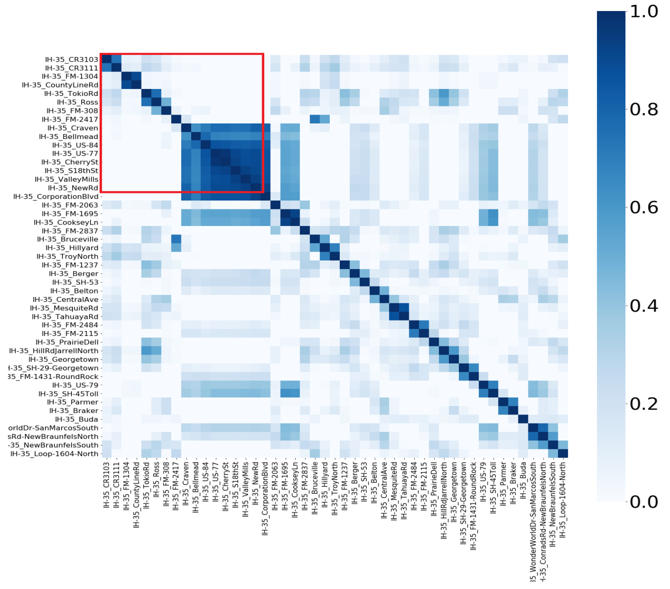

Figure 3. Correlation matrix of sensor pairs with zero time lag. The tick labels follow the spatial order of northbound sensors from bottom to top and right to left. Nodes with values in the red square (poor-performing) were omitted.

Notably, about 20% of our data was missing (with large spatiotemporal clustering; candidate reasons are examined in (V)). Noting the large cluster of high correlation coefficient values north of the "IH_35-CorporationBlvd", it was shown that these poor-performing nodes failed in tandem for a significant part of the year. Nodes north of this point were removed. This resulted in 32 speed and volume readings every five minutes, roughly 3.3M data points in total.

III. METHOD

To predict the future traffic flow, we implemented a simplified version of the first-order approximation model described in [3]. The specific model implementation was inspired by [5]. The model consists of so called ST-Conv blocks (Spatial Temporal Convolution). These blocks are made up of temporal and spatial convolutions. The temporal convolution is a 1D convolution done for each node in isolation from the others, only the data in that node across time is used. This convolution is done to two copies of the input data, one of these is then put through a sigmoid activation function. These two are then added and put through a final ReLu activation function. The temporal block consists of a feature matrix θ and the distance matrix D. The convolution is carried out by multiplying the data matrix X with the distance matrix, this picks out the values of the node and all nodes that are directly connected to it. These values are then multiplied by the feature matrix and then passed through a ReLU. A graphic representation of these blocks can be seen in fig. (4) and a graphic interpretation of how the convolutions are carried out can be seen in fig. (5).

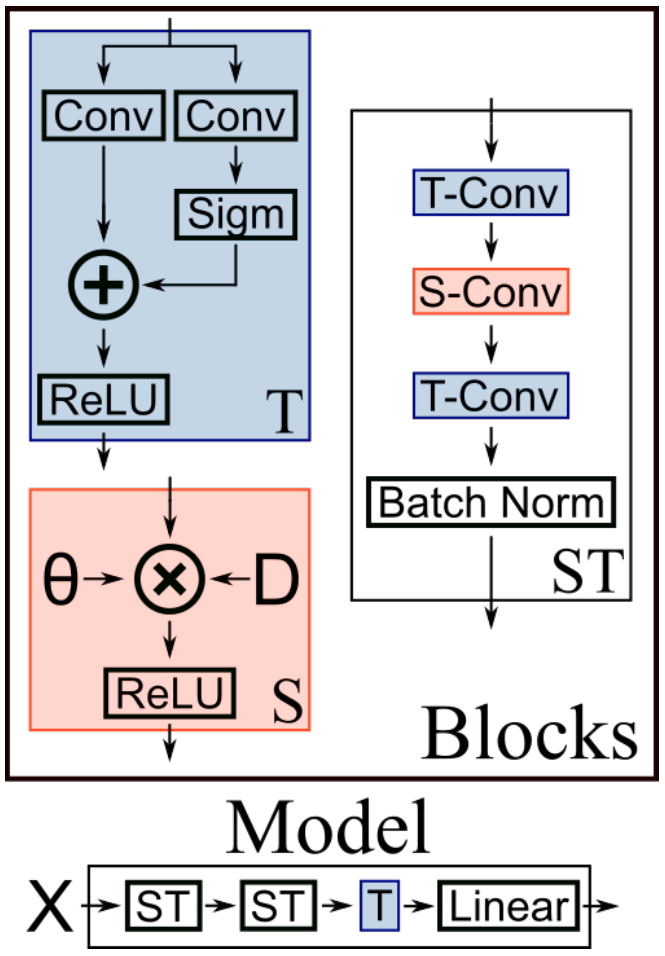

Figure 4. Visualization of the model and S, T and ST blocks which are used to build the complete model.

The decision to only make use of each node's direct neighbors in the spatial convolution was made by studying the correlation matrix of the sensors. It can be seen in fig. (3) that each sensor has a high correlation to only a few other sensors, ignoring the large, dark square (see Discussion). This informs us that the features at each sensor are mostly correlated with its direct neighbors. Seeing this, it seems reasonable to only make use of the direct connections in the spatial convolutions. For similar reasons, it is reasonable to expect correlation at each sensor to itself is the highest for near time steps. That is, we expect the traffic one time step back or forward in time to be more correlated to the current traffic as compared to the traffic far away in time. This, of course, does not take seasonality into account which this model was not made to understand as its predictions are on a short time scale (minutes to hours). For this reason, we chose to use a kernel size of 3 for the temporal convolutions.

These blocks are combined into a ST block which makes use of these and a batch normalization layer. The final model is produced by stacking two ST Blocks together, followed by a final Temporal block and lastly a fully connected layer. This gives the final model which can be seen in fig. (4).

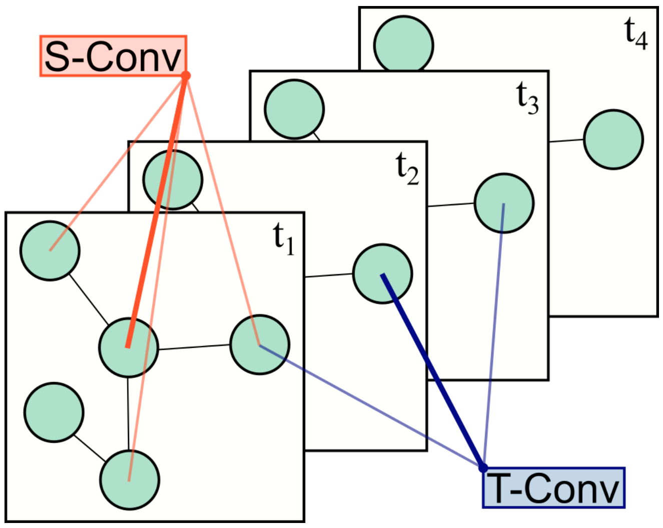

Figure 5. Graphic representation of data and the behaviour of the temporal and spatial convolutional blocks. The spatial block computes an output for each node at each time step using it and its direct neighbors, the temporal block computes an output for each node at each time step using only that node in the current time step and one step ahead and on behind.

Given the noisiness of the data set, an attempt to improve the predictive power of the model was made by introducing an extra time-periodic feature. We created three versions of the model, one trained on the data without any extra features, one with an added saw-tooth triangle wave, and a sin wave. Both periodic features had an amplitude of 1, a period of one day, and were always positive. These extra features can be seen together with some sample data in fig. (6). The saw-tooth wave was constructed such that it increases from 0 to 1 over the course of an entire day. The sine wave was constructed to go from a minima to a maximia and back to a minima over the course of an entire day. These three cases will be referred to as the normal, sine, and triangle case henceforth.

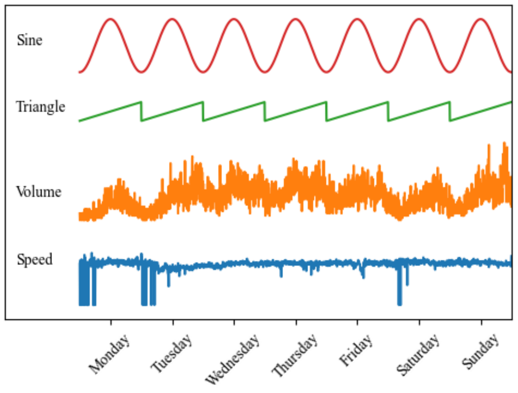

Figure 6. One weeks worth of data from a single sensor together with the proposed extra time periodic features. Note the constant nature of the speed data and the noisiness of the volume data.

The model was constructed using the PyTorch python package and trained using a holdout methodology where a total of 50 training epochs was allowed. The best performing model in regards to the validation data MAE was selected. The training was done on a RTX 2080 Ti and took roughly 20 min to complete. The complete code for the data prepossessing and the ST-Conv model can be seen in our GitHub page [6].

IV. RESULTS

The table below shows the test Mean Absolute Error scores for the two models together with a simple averaging scheme to predict the future data points. The models were trained on two different data splittings.

TABLE I. MAE test scores for the two models and the averaging scheme for two different data splits.

Train/Val/Test	ST-Conv	ARMA(7,20)	Averaging
65/20/15	5.09	3.73	3.06
90/5/5	4.20	4.24	3.76

The following figure shows the traffic speed averaged over every sensor, a metric representative of the average congestion along the highway. We see that the model can predict the average speed within several mph, as can also be seen in the MAE scores in the table above.

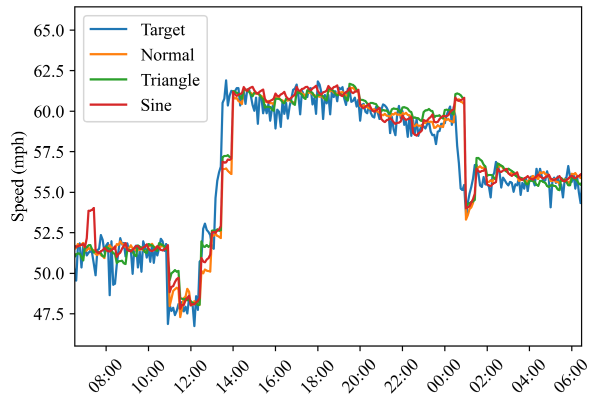

Figure 7. I-35 speed predictions averaged across every sensor in the network for the ST-Conv model. The predictions are made using 1 hour of previous actual data to predict the following 30 min. Each prediction is based on actual data and not using previously predicted data.

In the figure below we see the validation MAE scores of the ST-Conv model for the different approaches.

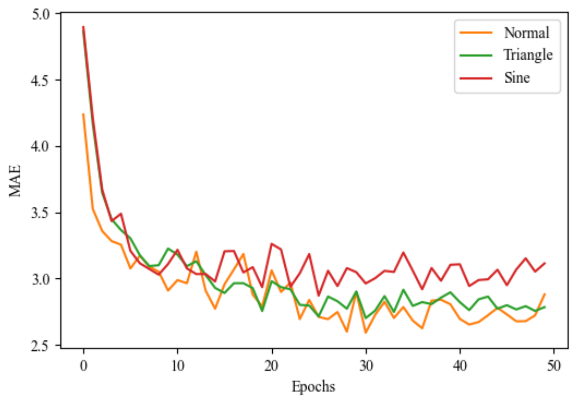

Figure 8. The MAE of the validation data versus training epoch.

V. DISCUSSION

There are a number of limitations to the Bluetooth data collection method, not limited to device and communications failures, road maintenance, and crashes. An inherent limitation is that traffic at a standstill and an empty road will look identical in the dataset, as speed is calculated from the arrival times and distance between two sensors. All of these errors appear as -1 in our dataset. One notable incident was the concurrent and lasting failure of sensors around Waco, TX, resulting in high correlation coefficients among these sensors (in Figure 3, between IH-35_Craven and IH-35_CorporationBlvd).

In fig. (7) we see that the model is capable of predicting the traffic speed within a couple of miles per hour. On average, it does so with an error rate of roughly 4 mph, depending on how many sensors are malfunctioning. To compare our results, we make use of a commonly used model for traffic predictions, the AutoRegressive Moving Average. A notable disadvantage of this model is the lack of spatial awareness inherent to the ST-Conv model. We fit ARMA models to each sensor and used the Bayesian and (corrected) Akaike Information Criterion, as well as test MAE, to determine the most suitable values for p and q. We employed $ARMA(7, 20)$ taking into consideration hardware constraints imposed by the size of data and number of models to fit. Looking at the MAE scores in table (I), we see that for the split with less training data, the ARMA model performed much better. During training, we saw that the ST-Conv model did not overfit on the training data, as can be seen in fig. (8). This led us to believe that we required more training data. To test this, we constructed the second split with a higher amount of training data. As can be seen in the table, this improved the performance of the ST-Conv model and made it comparable to the ARMA model.

However, a question that arises is how good are these predictions? We can make use of the simple method of averaging to help get an intuitive feel for the performance of the model. The method takes the last hour of speed data and computes the average to predict the following half-hour at each sensor. When computing the MAE using the same test data, we find that this model outperforms the two other models on both data splits. This hints at an underlying issue, the data itself.

The stretch of the I-35 used in this project connects two rural areas with relatively low populations. This in turn means that the I-35 is rarely above capacity, meaning that the average traffic speed is nearly constant. This can be seen in fig. (6) where the speed data is nearly constant for a week. Together with the noisiness of the data and the missing values, any kind of daily trend is only made evident when averaging over all sensors which can be seen in fig. (7). Since this is the case, the spatial relations between the sensors diminish in importance, as the trends we are trying to predict only appear once the individual sensors have been averaged away. With this in mind, it is understandable why the ARMA model performed equally or better as the ST-Conv model's only advantage, spatial awareness, has lost most of its significance. As to why the ARMA model is outperformed by averaging is likely the missing data since the averaging model gets a near perfect MAE score for these regions.

One possible way to improve the data issue would be to apply a stratified scheme to the training data. As can be seen in fig. (7), there exists a couple of regions with roughly constant speed separated by transitions. The issue with the training is that the constant regions far outweigh the transitional regions, which is made even more clear when studying the none averaged data for each sensor that is used when training. For this data set, the regions of interest to predict are these transitions and therefore the model would likely improve if these regions and the constant regions were equally represented in the training data.

Another method that we implemented in an attempt to remedy the data issue was the addition of extra time-periodic features. The underlying thought behind these was to help the model associate the connection between the time of day and the transitions. However, we found that the addition of these features did not improve performance. We believe that this was due to the limited amount of training data. As was seen in the test MAE table (I), the ST-Conv model improved significantly with more data, this is made even more evident as the averaging scheme performed much worse on the test data in this region, signifying more noise. The addition of extra features also entails more trainable parameters in the model requiring more data to train.

The implications of this study are unfortunately few, but can be summarised as such. We have found that the ST-Conv model is more resistant to real world datasets that contain missing values, as the ST-Conv model was able to produce results comparable to the averaging method on the original dataset, whilst the ARMA model required a much narrower purged dataset to function. This is due to the fact that the ST-Conv model utilizes convolutions, thus if a single senor malfunctions the convolution step can still utilize the sensors neighbors to produce a reasonable value. The ARMA model, however, works on each sensor in isolation, which causes it to suffer harshly when encountering missing values. The trade off for this improvement is the requirement of additional data to train the model. To further test the performance of our model, which was a simplified version of the model described in [3], a larger traffic dataset from a region with more congestion and preferable more road interconnections, such as a city, would be required.

VI. CONCLUSIONS

In this paper, we use the deep learning model STGCN to perform traffic prediction on the I-35 road in Texas, where we make use of both spatial and temporal information. Experiments show that both the standard model ARMA and simply averaging outperforms our model on this particular dataset. This indicates that models which make use of spatial information may not be suited on datasets where the average traffic speed is nearly constant.

Furthermore, we found that the use of convolution in our model makes it more robust to missing values, capable of producing results where the ARMA model would fail.

VII. ACKNOWLEDGMENTS

Data and thoughtful insight were provided by Robert Brydia of the Texas A&M Transportation Institute.

VIII. CONTRIBUTIONS

The data was prepossessed and analysed by Mathias and Colton. The ST-Conv model was implemented by Mathias and the ARMA model by Colton. The Abstract and Introduction was written by Oskar, the Dataset section by Colton, the Method section by Mathias and Colton, the Results section by Mathias, and the Discussion section by Mathias and Colton. In general all authors provided corrections and re-phrasings of the others sections.

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, Short-term traffic forecasting: Where we are and where we're going, Transportation Research Part C: Emerging Technologies **43**, 3 (2014), special Issue on Short-term Traffic Flow Forecasting.

[2] T. Zhang, Y. Liu, Z. Cui, J. Leng, W. Xie, and L. Zhang, Short-term traffic congestion forecasting using attention-based long short-term memory recurrent neural network, in *International Conference on Computational Science* (Springer, 2019) pp. 304–314.

[3] B. Yu, H. Yin, and Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence 10.24963/ijcai.2018/505 (2018).

[4] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, Forecasting traffic congestion using arima modeling, in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)* (2019) pp. 1227–1232.

[5] F. Opolka, Stgcn-pytorch, `https://github.com/FelixOpolka/STGCN-PyTorch` (2018).

[6] M. Toftås, Ml-w-ann, `https://github.com/cunov/ML-w-ANN/` (2021).