**Ziyin Wang, Austin Bell, and Malik Drabla**

**Milestone Report**

**Outline**

* Introduction
  + Project overview
* Progress
  + Summary of progress
  + Data Processing
    - Time series sequences
    - Normalized adjacency matrix with self-loops
  + Models
    - DGL GCN
    - Baseline STGCN
* Evaluation
  + Model results MAE/MSE/RMSE
    - DGL
    - STGCN
  + Plots of predictions vs. gold standard (STGCN)
* Discussion
  + These are not good initial results
  + Therefore, we are focusing on STGCN that have a larger temporal component
    - STG2seq
      * Spatial temporal graph with seq2seq architecture
  + Incorporate forecasts and metadata
  + Additionally, a decreasing learning rate seems to be an important factor

**Report**

1. **Introduction**

Since the industrial revolution, our earth has been warming due to the release greenhouse gases from burning ‘dirty’ fuel sources. A recent surge in clean energy investment has led to more widespread adoption of renewable, clean energy, however we note that there still remains many technical barriers to its adoption1. In particular, we highlight two technical:

* Geographic siting of plants: Fossil fuel plants are built remotely and energy production is centralized, whereas, renewable energy plants are decentralized and sometimes built in residential areas
* Storage or excess energy: the production of renewable energy is unpredictable leading to less efﬁcient allocation of energy supply

While a wide net of solutions have been proposed and are currently being pursued, our project focuses on the use of AI to improve efﬁciency in the allocation of energy. We propose using spatial-temporal graph neural networks to accurately forecast load for each decentralized node. Through better predictions of energy demand, we can partially address each of these technical barriers.

* Spatial analysis enables us to better capture any geographic dependent features related demand (e.g., nodes providing electricity to cities are likely to be characterized by more extremes)
* Proximity of nodes will likely affect demand as one node reaches capacity they may divert the provision of energy to the nearest node that is below capacity

Overall, by better understanding our exact energy needs, we will be able to better optimize power ﬂow across decentralized plants, ensure supply more closely meets demand, and thereby, lessening our reliance on advanced storage solutions

1. **Methodology**

The goal of our project is to improve forecast prediction of load demand for each node (i.e., power provider) through leveraging Spatial Temporal Graph Neural Networks (“STGNNs”). Our power provider nodes will represent our graph nodes and their transmission connections will represent the edges. Each node includes a variety of data at each timestamp such as hourly demand, wind / solar forecasts, and wind / solar realizations.

A spatial temporal graph neural network is a small expansion on the original graph neural networks. Where, traditionally, graph neural networks only consisted of a spatial component (i.e., graph convolutions on an adjacency matrix), a STGNN adds a temporal component. Current, temporal models include RNNs, Transformers, but also temporal convolutions. As of now, temporal convolutions are typically used in STGNNs due to their speed. We use these spatial and temporal convolutions at each time sequence (e.g., 24 hour historical load and solar data) to effectively forecast our time series.

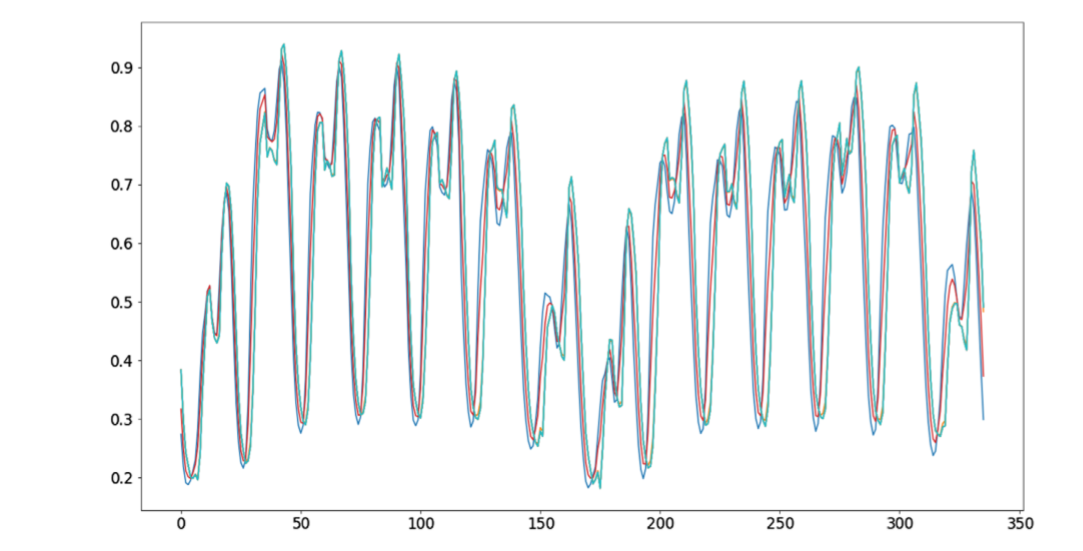
Our baseline model will be an implementation of the spatial temporal graph convolution network (“STGCN”) originally developed by Bing Yu and his team in 2018[[1]](#endnote-1). Subsequent models will be selected based upon perceived weaknesses of the baseline STGCN. For example, would the model benefit from additional temporal layers or would the model benefit from additional spatial layers. Once we are able to identify these ‘problem areas’, we will explore the current state of the art that best fits our problem. We currently are unaware of any other paper that uses graph networks to predict energy demand, which means we will likely need to evaluate the effectiveness of STGNNs ourselves.

Experiments are run to forecast day ahead demand. For each experiment, we evaluate the time series output using mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE).

1. **Dataset**

We utilize the recently developed RE-Europe dataset which consists of nodal load demand across mainland Europe and also includes a transmission network model, plus data regarding electric generation and demand from 2012-2014.[[2]](#endnote-2) There are 1,494 different nodes with 2,156 transmission connections. Typically, load demand at the node level is highly confidential data, therefore, there are no other datasets including this granular of information.

The below figure displays hourly demand that has been normalized to fit between 0 and 1 across three similar nodes:



*Figure X: Normalized Load Demand at Hourly Level across three nodes*

*Source: Multi scale deep network based multistep prediction of high-dimensional time series from power transmission systems*

1. **Summary of Progress**

Our team has made significant progress in our project so far. The first milestone was aimed at setting our baseline and implementing a (hopefully) near state of the art solution. Whereas, the following half of the project will be focused on implementing and expanding upon current state of the art techniques within STGNNs. Below, we list a summary of our current progress:

* Processed, standardized, and prepared the data for spatial temporal forecasting
* Developed a DGL Graph Convolution Network that ignores the longitudinal component
* Implemented the baseline Spatial Temporal Graph Convolution Network

The following sections details our process and methodologies employed in each step.

* 1. **Data Processing**

Our first methodology involved taking the provided data and reformatting it to work with our spatial temporal GNN. Our initial baseline model just takes a graph and a tensor of our longitudinal features. Our features are represented via a 4D tensor with dimensions: (batch number, number of nodes, historical timesteps, and number of longitudinal features).

Prior to any processing, we focused on doing some general feature engineering given the data that we had. This included identifying the day of the week (whether it was a weekend), identifying the month, identifying the season, and identifying whether or not a holiday occurred on that date. Figuring out whether a holiday occurred required slightly more work as our data is based across all of Europe, where each country has their own set of holidays.

While processing the data, we realized that there were a lot of different methodologies that we could pursue, therefore, we focused on developing robust enough data processing code that enables us to rapidly experiment with a variety of input data. This includes using different methods to normalize our data, utilizing a different number of historical timesteps as our input, and including different sets of features in our model.

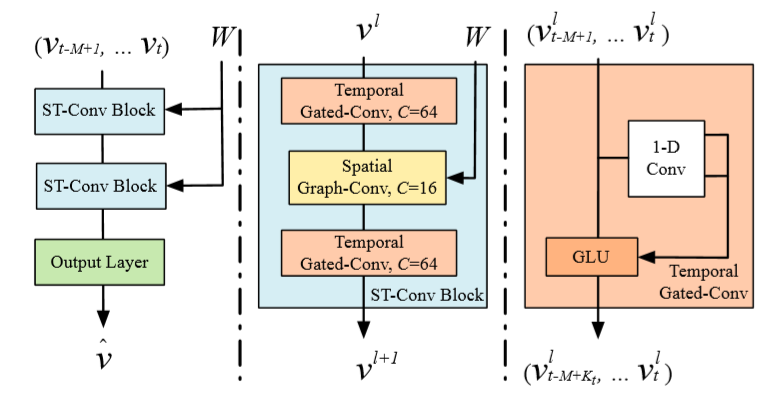
Finally, our baseline model is written in pure Pytorch and excludes Pytorch Geometric. This means that we are unable to pass a graph into the network and must instead utilize our graph’s adjacency matrix. In order to effectively utilize our adjacency matrix, two processing steps were required: 1) normalizing it with the formula D^(-1/2)AD^(-1/2), where A is our adjacency matrix and D is our degree matrix and 2) adding self-loops to ensure that we include the current node and not just those in the neighbourhood.

* 1. **DGL Graph Convolution Network**

[[Ziyin to add]]

* 1. **Spatial Temporal Graph Convolution**

As mentioned our baseline model is based on the original STGCN. This model consists of a series stacked temporal and spatial layers. This includes two STGCN blocks, a final temporal convolution, and a output linear layer. Each STGCN block consists of a temporal convolution layer, a spatial graph convolution, and another temporal convolution layer. Temporal convolutions were selected over sequence or attention-based models due to their speed. Figure X below shows a picture of this process:



*Figure X: STGCN Architecture*

*Source: Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Trafﬁc Forecasting*

For the baseline, we train for up to 200 epochs (with allowed early stopping) at a learning rate of .001 that decreases every 30 epochs if the model is not improving.

1. **Evaluation**

Next, our team evaluated the results of each of these models. We evaluated across all available nodes in our dataset on the last three months in our data (October 1st, 2014 through December 31st, 2014).

* 1. **DGL Evaluation**

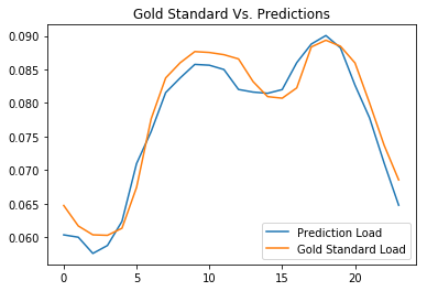
[[Ziyin to Add – actual MSE loss numbers may not make sense here? Maybe just a summary? Idk, however you would like]]

* 1. **STGCN Evaluation**

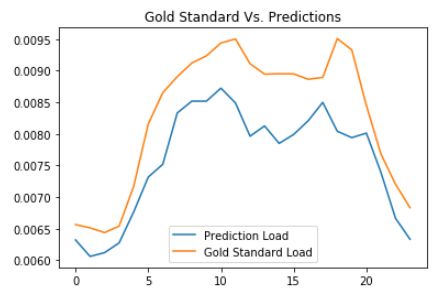
The below table compares our model across three metrics to the current state of the art using the RE-Europe dataset for load demand forecasting. We note that this is also the only paper that we have identified leveraging this dataset for spatial temporal prediction. Their model utilizes a multi-scale network combined with a seq2seq network[[3]](#endnote-3).

|  |  |  |  |
| --- | --- | --- | --- |
|  | **MAE** | **MSE** | **RMSE** |
| **STGCN** | .0038 | 6.62e-5 | .0081 |
| Multi-Scale + Seq2Seq | N/A | 4.92e-7 | N/A |

As can be seen, our initial results are less than stellar. It appears that our model is learning to predict some nodes well, yet some nodes not at all. Two examples are provided below:



*Figure X: Node 1 random 24 hour load forecast*



*Figure X: Node 100 random 24 hour load forecast*

1. **Discussion**

Our current hypothesis for these poor results is that STGCNs are typically used for forecasting traffic, which does not have the same level of temporal dependencies as day ahead load forecasting. Typically, load forecasting requires much longer historical timestep inputs and must also predict much longer forecasts. Therefore, we believe that the basic temporal convolution is not sufficient for our problem.

Therefore, our next step is to implement a spatial temporal graph neural network that includes a sequence model for the temporal component such as the one developed by Bai et al (2019)[[4]](#endnote-4). Their team combines the spatial temporal graph with a seq2seq architecture with attention to better model passenger demand for ride hailing services in China. We believe that this seq2seq architecture will rapidly improve our models’ effectiveness.

Additionally, we believe that our current model is missing two key pieces of available data. The first is that we are excluding our solar and wind forecasts since we only include historical data at the moment. However, we expect that including weather indicators would signify whether people stay indoors and use more energy or go outdoors. The second piece is that we are excluding the historical and current day metadata. This includes whether the day is a holiday, which season, etc. We excluded this because it decreased our results when including as a traditional time series. However, past research shows that by incorporating metadata as cross-sectional data after a linear layer leads to improved results.

Finally, based on our recent experiments, it appears that tuning our hyperparameters will be quite import. First, small changes in the learning rate has had big impacts on our model’s outcome and finding the optimal rate has proved elusive. Our team will work on tuning the learning rate while also implementing a more sophisticated learning rate scheduler. Secondly, we trained our model for 200 epochs and the model was still improving slowly. The current state of the art trained for around 800 epochs. We will ultimately train for this long, but will wait until we have a more complete model.

1. **Conclusion**

Despite the poor original results, we are quite happy with the progress that our team has made so far. We have fleshed out our methodology, processed the data, and fully implemented then evaluated our two baseline models. Additionally, since this is a longer term project, we have implemented a robust framework that will now allow for rapid and efficient deep learning experimenting. We plan to efficiently use the next month to explore and, subsequently, surpass the current state of the art.

1. Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In IJCAI. [↑](#endnote-ref-1)
2. Jensen, T., Pinson, P. RE-Europe, a large-scale dataset for modeling a highly renewable European electricity system. Sci Data 4, 170175 (2017). [↑](#endnote-ref-2)
3. Zhu H, Zhu Y, Wang H, et al. Multi scale deep network based multistep prediction of high-dimensional time series from power transmission systems. Trans Emerging Tel Tech. 2020; e3890. https://doi.org/10.1002/ett.3890 [↑](#endnote-ref-3)
4. Bai, L.; Yao, L.; Kanhere, S.; Wang, X.; and Sheng, Q. 2019. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. In IJCAI [↑](#endnote-ref-4)