

# Graph Convolution Networks for Probabilistic Modeling of Driving Acceleration

박종철 Jongcherl Park

Seoul National University

Term Project Final Report

# 드리는 말씀

기존에 교수님께는 이메일로 말씀드렸지만, 건강상의 이유로 본 강좌 수강을 소홀히 하였고 최종 동영상 제출하지 못하였습니다. 다시 한 번 죄송하다는 말씀 교수님께 드립니다.

선택한 기존 논문에 대해 주어진 Source code가 없어 제 힘으로 코딩을 해보려고 했지만, 기존에 주로 matlab 및 C로 코딩을 해왔던지라 python에 익숙해지는데 시간이 걸렸고, 사실 이번 수업이 제가 Neural Net에 대한 기초만 알고 Graph Theory는 본 강좌에서 처음 배웠으며, Tensorflow를 다룬 경험이 매우 적다 보니 간단한 GCN sample code를 공부하는 데에도 굉장히 오랜 시간이 걸렸습니다.

때문에 Source code 없이는 논문 재현이나 GCN을 사용한 code를 작성하기 어렵다고 생각하여, 기존에 발표했던 논문이 아니지만 비슷한 주제를 다루고 있고, 기존에 다뤘던 논문과 유사한 방법을 사용한 다른 reference 논문을 찾았습니다. 해당 논문은 Source code가 github에 open되어 있어 제가 쉽게 찾아보고 따라해 볼 수 있었습니다.

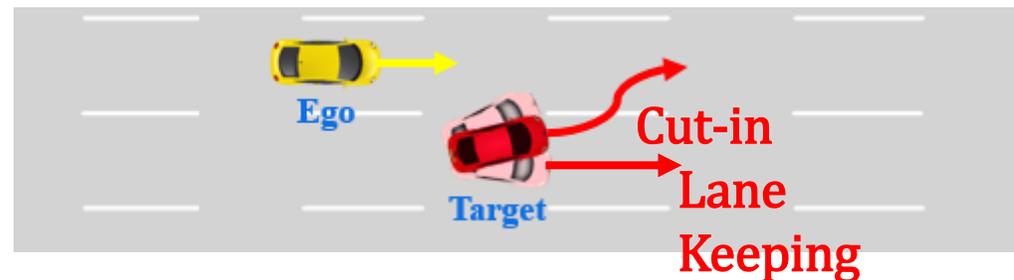
비록 Tensorflow 및 GCN에 대한 이해가 부족하여 source code를 완전히 이해하진 못하였지만, 논문의 내용이 실제 code에 어떻게 구현되어있는지 조금은 살펴볼 수 있었습니다. 해당 논문에 대한 간단한 summary 및 source code 요약 내용은 기존에 review했던 논문 발표자료 뒤에 첨부하였으며, sample data에 대한 실험결과도 첨부하였습니다.

제 능력이 부족하여 Term project에서 요구했던 내용을 거의 수행하지 못했습니다. 죄송합니다.

실행 코드의 경우 데이터와 함께 첨부하였으며, 제가 따로 수정한 것은 없고, 몇 가지 주석만 달아 놓았습니다.

# Target Prediction Task in Autonomous Vehicle

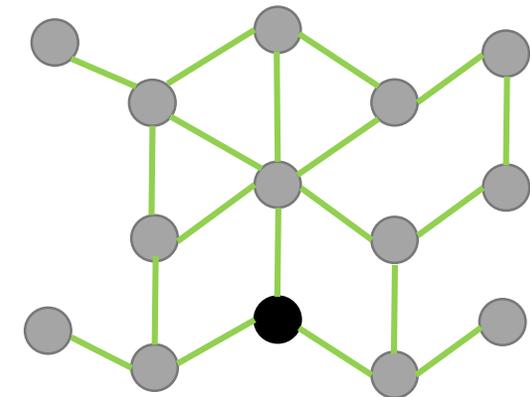
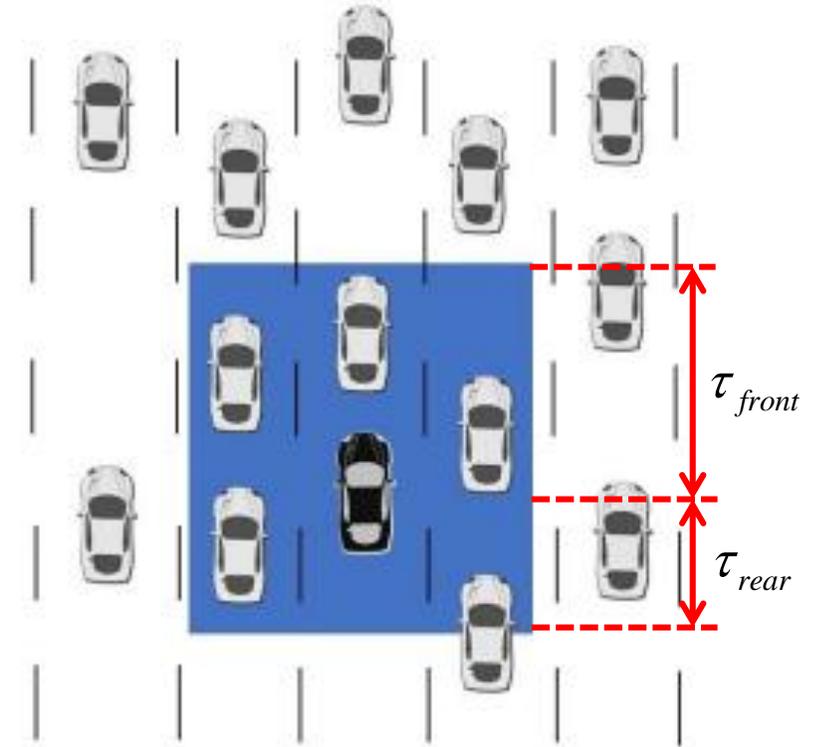
- Longitudinal Prediction (Cruise Control, Emergency Brake)
- Lateral Prediction (Cut-in, Cut-out)



# Traffic graph (1/2)

- Vehicles interact with each other
- Each node represents each vehicle in traffic
- Two nodes are connected to each other with edge if they meet 2 conditions:
  - They are in same or adjacent lane at the same frame
  - Longitudinal distance is less than threshold  $\tau$

Traffic Graphs



## Traffic graph (2/2)

- Feature vector of each node contains:
  - lane id, vehicle class, vehicle velocity, vehicle acceleration,
  - relative longitudinal distance from 3 nearest front vehicles
  - relative longitudinal distance from 3 nearest rear vehicles
- NGSIM I-80 dataset
  - Vision based large scale traffic dataset
  - 10fps – each data represents 0.1s information
  - Average 110 vehicles per single video frame (=110 nodes)
  - Average 4 edges per single node (=4 neighboring vehicles)

# GCN Structure

- Distance-aware GCN

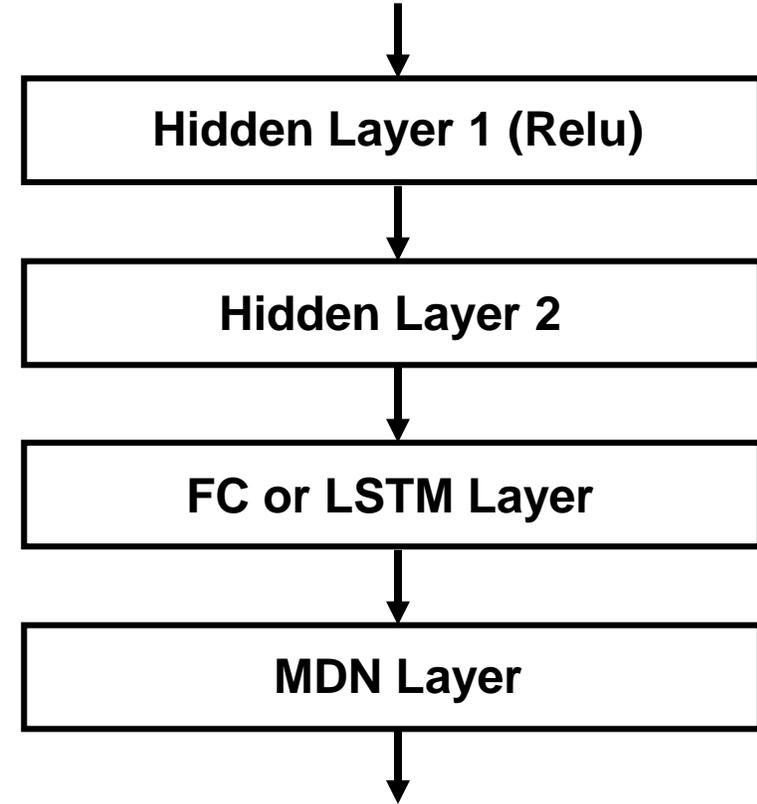
$$H^{l+1} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l + I_N H^l B^l \right)$$

$$\text{with } \tilde{A}_{ij} = \frac{1}{|y_{v_i} - y_{v_j}|}$$

$$h_{v_i}^{l+1} = \sigma \left( \sum_j \frac{h_{v_j}^l}{\tilde{c}_{ij}} W^l + h_{v_i}^l B^l \right)$$

- Mixture Density Network

$$p(a) = \sum_{i=1}^K w_i \mathcal{N}(a | \mu_i, \sigma_i^2)$$



# GCN Structure

- Baselines

Proposed models are compared with non-GNN models and GAT models

Each model has extra layer of either FC or LSTM

Model	layer 1	layer 2	layer 3	MDN layer	LSTM	clip norm	adjacency type
Fully-connected	128	256	128	90	no	5	/
GCN base	128	256	128	90	no	5	binary
GAT	128	256	128	90	no	5	binary
EGCN	128	256	128	90	no	5	binary
DGCN	128	256	128	90	no	5	inverse distance
LSTM	128	256	128	90	yes	5	/
GCN with LSTM	128	256	128	90	yes	5	binary
GAT with LSTM	128	256	128	90	yes	5	binary
EGCN with LSTM	128	256	128	90	yes	5	binary
DGCN with LSTM	128	256	128	90	yes	5	inverse distance

# Evaluation

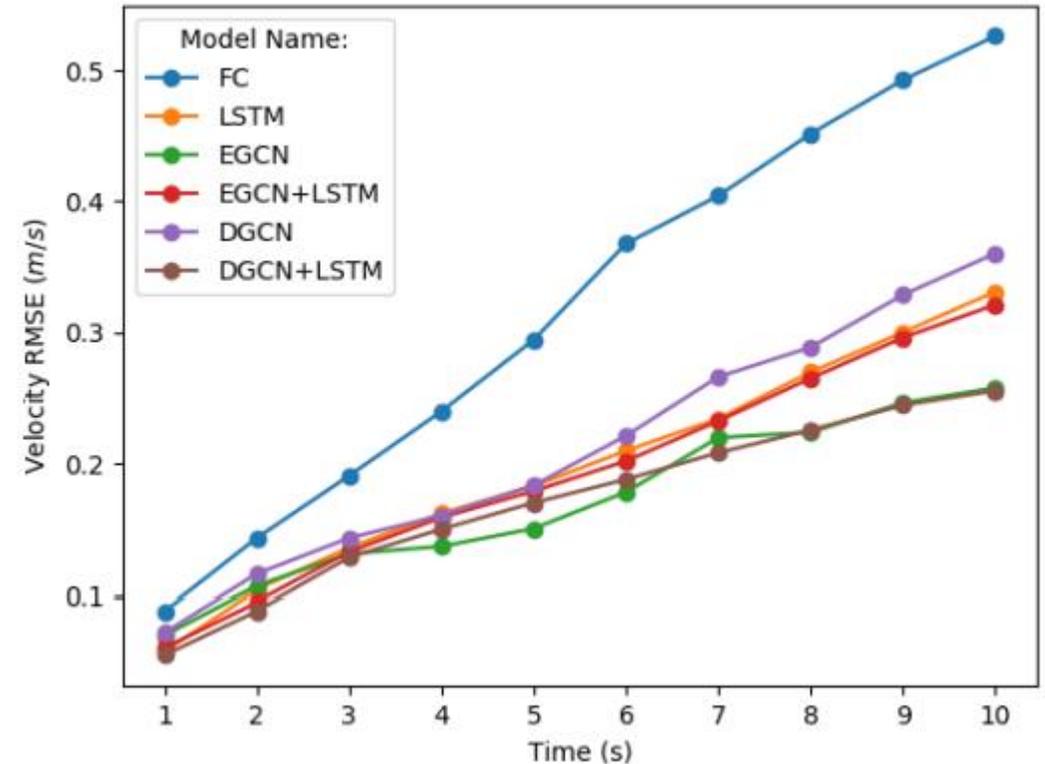
- RMSE error results outperforms other algorithms

- Prediction Horizon : ~10s

$$RMSE_{velocity} = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (v_H^i - \hat{v}_H^{i,j})^2}$$

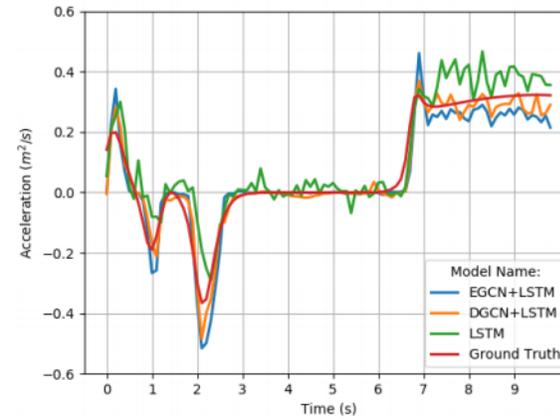
TABLE II: RMSE Analysis

Model	Y RMSE @ 10 s (m)	Velocity RMSE @ 10 s (m/s)
Fully-connected (FC)	2.89	0.526
GCN base	3.52	0.622
GAT	4.13	0.688
EGCN	<b>1.40</b>	<b>0.258</b>
DGCN	1.91	0.360
LSTM	<b>1.61</b>	0.331
GCN with LSTM	3.40	0.653
GAT with LSTM	4.09	0.728
EGCN with LSTM	1.86	0.321
DGCN with LSTM	1.63	<b>0.256</b>

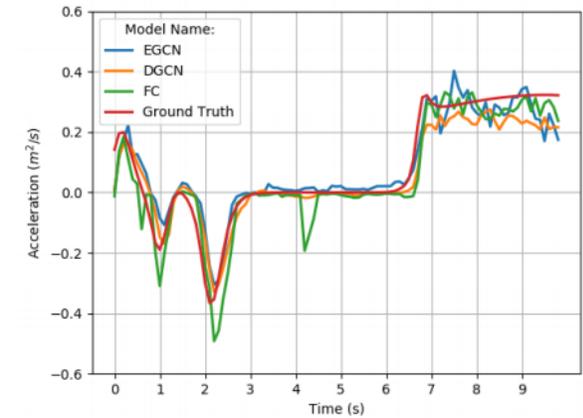


# Evaluation

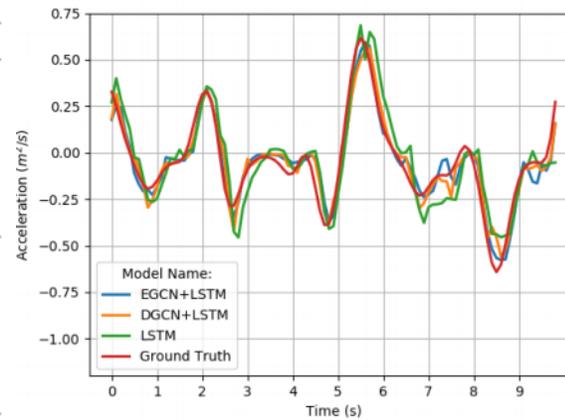
- Jerk Sign Inversions
  - Represents smoothness
- Negative Headway Occurrence Rate
  - Represents robustness



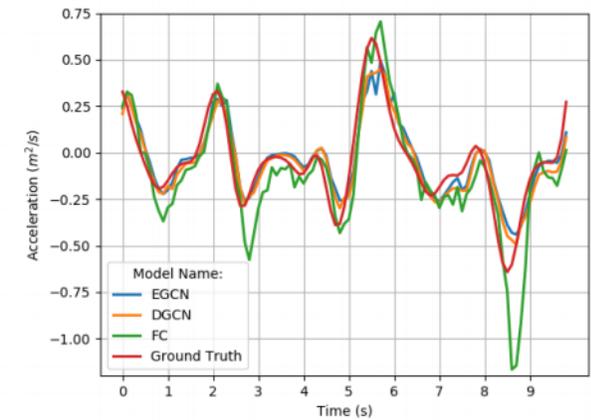
(a) LSTM models



(b) Fully-connected models



(c) LSTM models



(d) Fully-connected models

Model	Jerk Sign Inversions	Negative Headway Occurrence Rate
Fully-connected (FC)	7.5	0.08
GCN base	7.5	0.17
GAT	<b>5.9</b>	0.27
EGCN	7.5	<b>0</b>
DGCN	7.3	0.03
LSTM	13.7	0.02
GCN with LSTM	<b>6.7</b>	0.17
GAT with LSTM	0.0	0.27
EGCN with LSTM	9.5	0.01
DGCN with LSTM	7.3	<b>0</b>
True trajectory	6.3	/

# Conclusion

- GCN improves target prediction performance
- (May) considers interaction between vehicles
- Better results in position, velocity prediction
- Unreasonable prediction(which is critical to motion planning) less occurs

# GCN 및 Source 코드 학습을 위한 유사논문 탐색

## ▪ T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction

- <https://arxiv.org/abs/1811.05320>

- ITS 2019 발표 논문
- Traffic forecasting에 GCN과 GRU를 적용한 T-GCN을 도입
- 기존 논문의 acceleration forecasting에 GCN과 LSTM을 적용한 것과 거의 유사한 방법으로 접근
- Traffic의 spatial feature를 GCN, temporal featur를 GRU를 통해 학습하는 것이 main idea
  
- 기존 논문이 도로 상의 각 차량을 node로 설정, 인접한 차량들을 edge로 연결하였다면, 본 논문에서는 road network  $G$ 를 graph로 표현하는 것을 목표로, 개별 도로를 node, 연결된 두 도로들 간에 binary edge를 설정하여서 spatial relationship이 graph와 직접적으로 대입되는 직관적인 traffic graph 설정이 가능함.
  
- Node의 feature vector는 traffic speed(도로 상의 차량들의 평균속도)로,  $t-n$ 부터  $t$ 시점까지의 정보를 바탕으로  $t+1$ 시점부터  $t+T$ 까지의 feature vector를 예측하는 것이 본 논문의 목표

# GCN 및 Source 코드 학습을 위한 유사논문 탐색

- 기존 논문과 동일하게 2-layer GCN을 사용하였음

$$f(X, A) = \sigma(\hat{A} \text{ReLU}(\hat{A} X W_0) W_1)$$

- 기존 논문과 동일하게 3번째 Layer에는 RNN cell을 사용(LSTM/GRU)

$$u_t = \sigma(W_u [f(A, X_t), h_{t-1}] + b_u)$$

$$r_t = \sigma(W_r [f(A, X_t), h_{t-1}] + b_r)$$

$$c_t = \tanh(W_c [f(A, X_t), (r_t * h_{t-1})] + b_c)$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t$$

- Overfitting 방지를 위해 regularization term을 포함한 loss function 정의

$$loss = \| Y_t - \hat{Y}_t \| + \lambda L_{reg}$$

# Source Code 학습내용

- Summary

- Source Code는 Tensorflow 기반으로 작성되었음 (<https://github.com/lehaifeng/T-GCN>)
- 실행파일 main.py 에서는 변수 선언 및 data load, 성능 비교를 위한 baseline중 하나를 택하여 학습/테스트/성능 분석을 실행함
- Baseline은 HA, ARIMA, SVR, GCN, GRU, T-GCN(GCN+GRU)의 6개로 구성되어 있음
- 본 Term project에서는 baseline 성능 비교는 하지 못하였고, T-GCN cell에 대해서만 학습/실행

- Data

- 데이터는 중국 Shenzhen의 taxi 주행경로에서 1달간 수집된 traffic speed가 주어져 있음(매 15분마다 기록됨)
- Adjacency matrix 같은 경우에는 각 도로간 연결성이 변화하지 않는 데이터이기 때문에 고정된 값으로 주어져 있음
- 총 156개의 도로를 사용 (156X156 adjacency matrix)

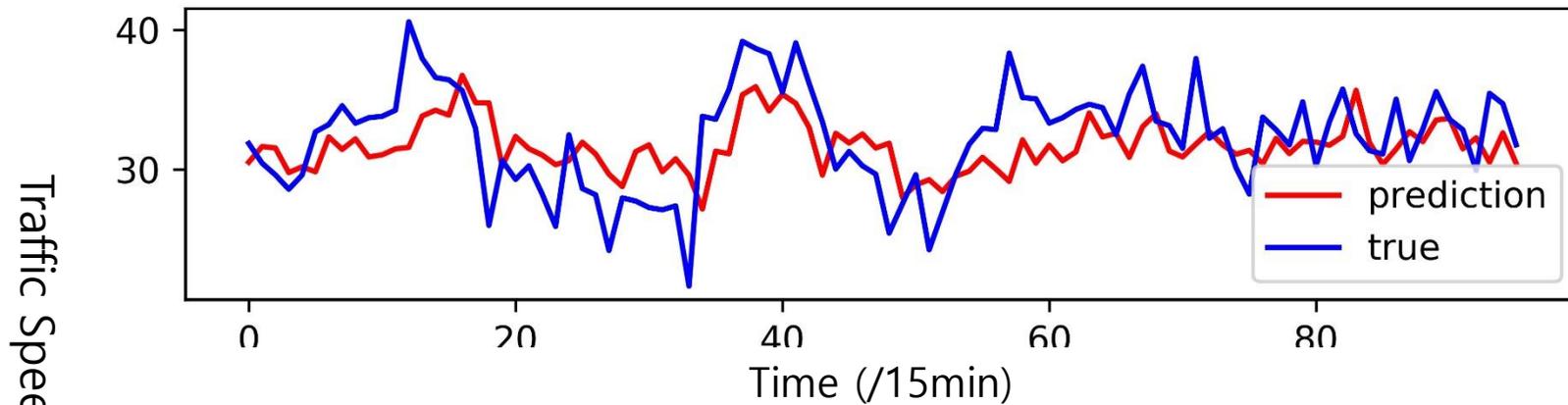
- T-GCN

- Single T-GCN cell은 두 개의 GCN Layer와 하나의 GRU layer로 구성되어 있음.
- Adjacency matrix A와 시간 t에서의 feature vector X(t)는 2개의 GCN Layer를 적용한 후, 이전 cell에서 가져온 GRU hidden state h(t-1)을 불러와 GRU layer를 적용하여 output Y(t)와 hidden state h(t)를 출력함

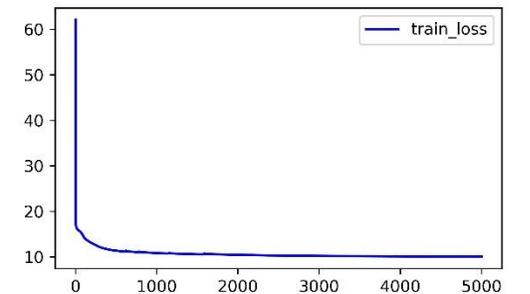
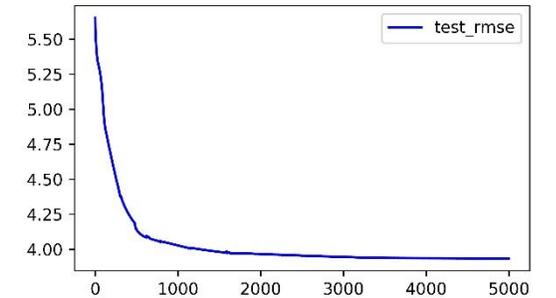
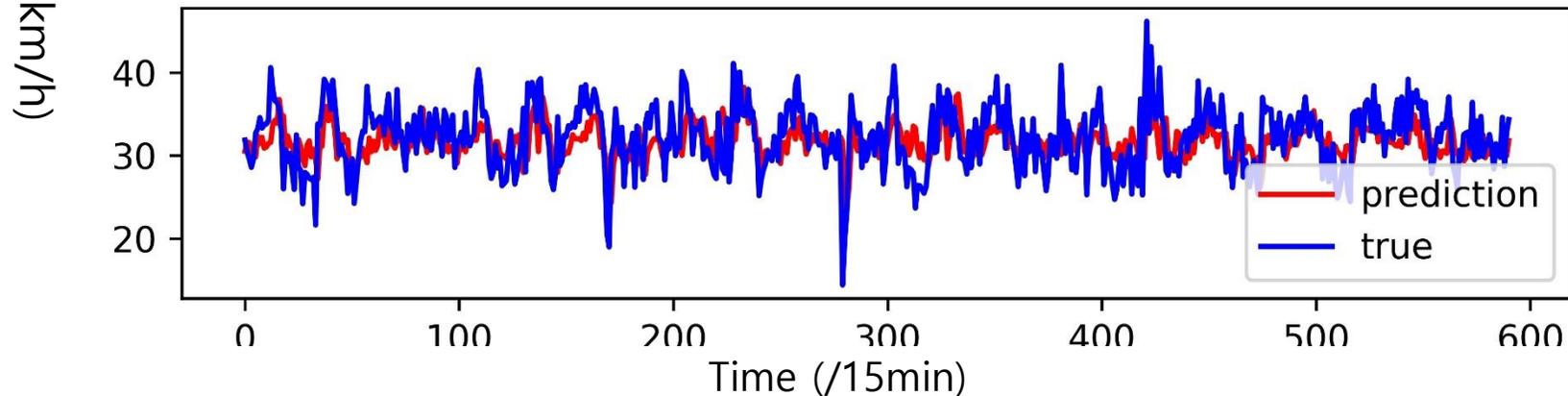
# Result

- Traffic Speed Prediction 결과 예시 (Single Road)

### One-day Result



### 1 month Result



감사합니다