AI Final Project Final Presentation
Team 11

Topic: Traffic prediction base on environment conditions and pedestrian demand

Group Name: Snot Cockroach and the Space-Time Rift
112550045 曾歆喬 112550047 徐瑋晨 112550099 蔡烝旭 112550190 劉彥廷

Introduction

In this project, we primarily focus on the automatic traffic flow in New York City. By training and utilizing AI models, considering some factors like pedestrian demand, weather, and historical record, to provide user with the prediction of approximate traffic volume on a given street, considering real time factors such as time of day, weather conditions, and other influencing variables to make the prediction more close to the real world conditions.

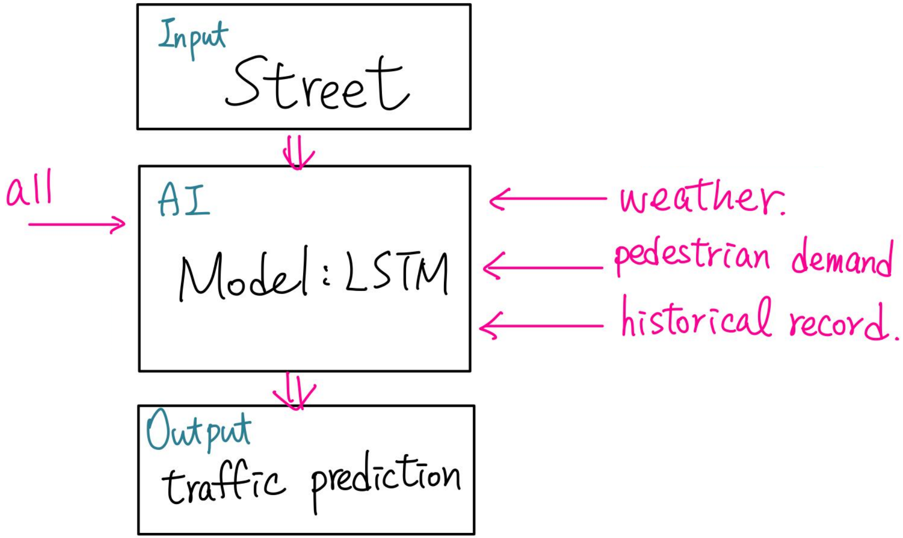
- **Weather conditions**: including rainfall, cloud cover, and temperature
- **Pedestrian demand**: represented by numerical labels
- **Air quality**: using historical PM2.5 measurements
- **Historical traffic patterns**: transformed into structured, analyzable data

Overall

Our AI model is built upon the Long Short-Term Memory (LSTM) architecture.

By incorporating three primary features, the model is trained to predict traffic levels for specific road inputs.

The red arrows we will explain on [Main Approach - input & output]



Related Work

Traffic flow prediction under multiple adverse weather based on self-attention mechanism and deep learning models Traffic flow prediction under multiple adverse weather based on self-attention mechanism and deep learning models

- In the previous studies, models were trained using historical traffic records alongside data from extreme weather conditions .
- In their project, they focused on highway and only use traffic and adverse weather data, (snowstorm, foggy…).
- Otherwise, our project focus on the **typical traffic flow on urban roads in NYC and daily weather condition.** In addition, we **incorporated pedestrian demand as a factor**, which has not been considered in previous work.

Datasets - Resource

- Traffic Volume Counts: the historical traffic volumn counts in the NYC.
- Automated-Traffic-Volume-Counts: another historical dataset of the historical traffic volume conuts in NYC.
- NYC Weather- 2016 to 2022 : the historical weather record of the NYC, including, temperature, rain, cloud cover…
- Pedestrian Mobility Plan Pedestrian Demand: The datasets including the pedestrian demand level (1 ~ 5), which is set based on the amount of pedestrian walking on the sidewalk of a specific road.
- Air-Quality: this dataset contains the recording of the daily air quality.

Datasets - Overview

- We combine the column we mentioned below:

  ID, Boro (Borough), Date, Hour, Weekday, Volumn (volume, It is a typo on our dataset), SegmentID, Street, Temperature, Precipitation, Rain, Cloudcover, Windspeed, Demand, Air Quality.

  The detail of dataset we will show on next page!

- We remove the street with fewer than 720 data entries.

Datasets - Details

- **ID:** Unique identifier for the record
- **Boro:** Borough or regional designation
- **Date:** Recorded date of the data point
- **Hour:** Time in hour
- **weekday:** weekday or not
- **volumn:** Traffic amount in each hour **(typo in dataset)**
- **segmentID:** Unique identifier for a street segment
- **street:** Name of the street
- **fromst:** Starting point of the segment
- **tost:** Ending point of the segment
- **Direction** Traffic or data collection direction
- **temperature:** Atmospheric temperature at the recorded time
- **precipitation:** Precipitation in mm
- **rain:** Precipitation only contain rain
- **cloudcover:** Percentage of cloud coverage
- **windspeed:** Speed of wind at the time
- **demand:** Pedestrian demand
- **Air_quality:** Measurement of pm2.5

Baseline - linear regression + trained by streets

To establish benchmark performance for our traffic volume prediction task, we implemented several **Linear Regression baseline models** by each street and help evaluate how different feature groupings influence prediction accuracy.

- **All Features : features = ['Hour', 'weekday', 'temperature', 'precipitation', 'rain', 'cloudcover', 'windspeed', 'demand', 'Air_quality']**

Baseline - linear regression + trained by streets

The method in baseline:
Using linear regresstion to build our baseline.

Some indexes:
We use RMSE, MAE, R^2 to evaluate.

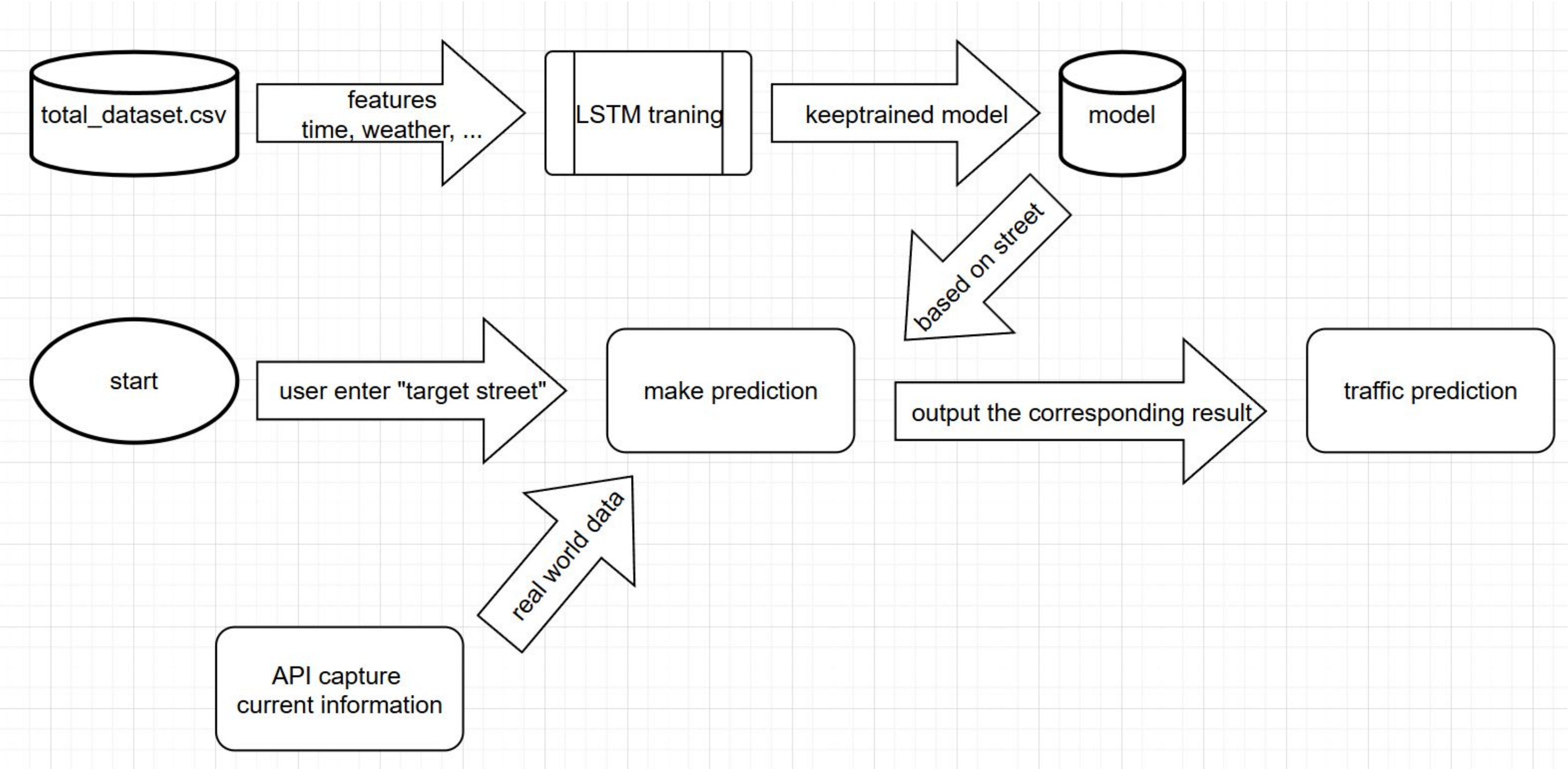
```
for street, group in df.groupby('street'):
    X = group[features]
    y = group[target]

    # 資料標準化
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # 訓練、測試集
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

    # 線性回歸
    model = LinearRegression()
    model.fit(X_train, y_train)

    # 預測
    y_pred = model.predict(X_test)

    # 指標
    r2 = r2_score(y_test, y_pred)
    rmse = root_mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
```

Main Approach - Overview

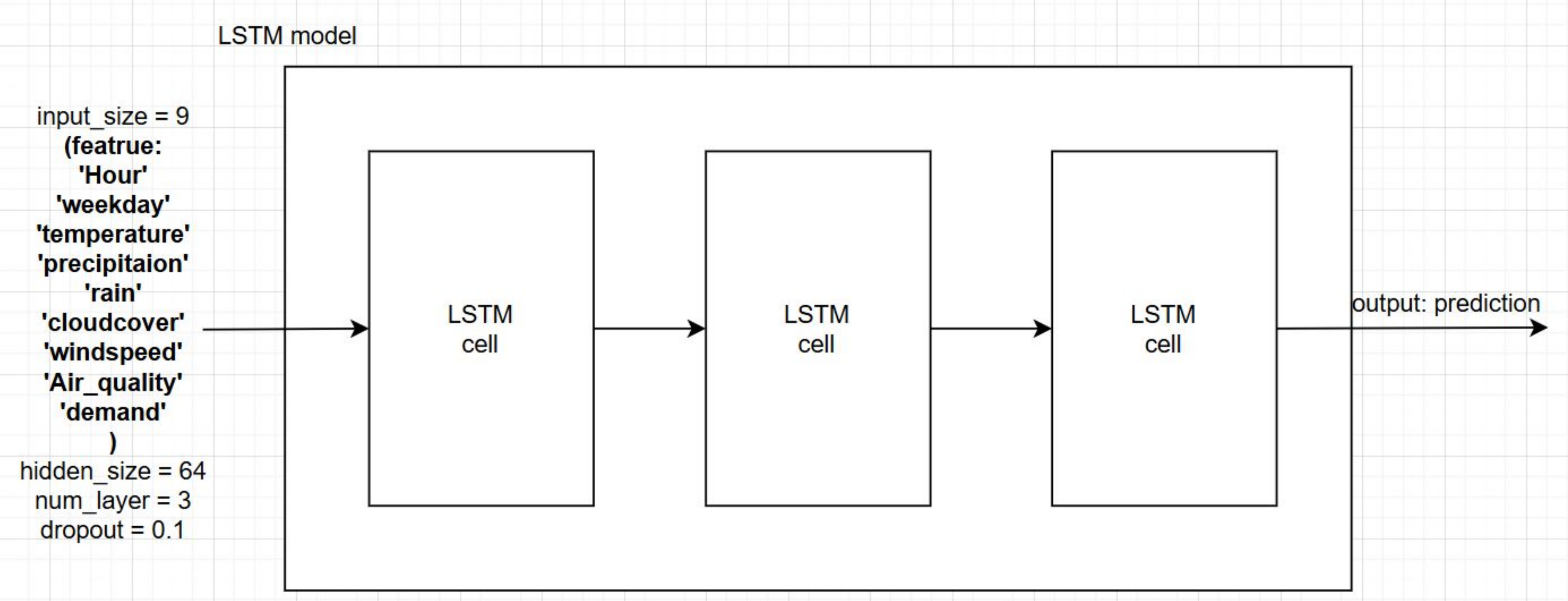
In this project, we choose using the model – "LSTM".

1. Before training model, we standardlized the input features and traffic volume saperately.
2. We separated dataset into two parts, 80% for training, and 20% for testing.
3. We used training dataset to train LSTM model follow the below diagram.
4. We tested our model with testing dataset.

Main Approach - Overview



Main Approach - Model based on LSTM architecture



Main Approach - LSTM Cell Overview

i = sigmoid(i)

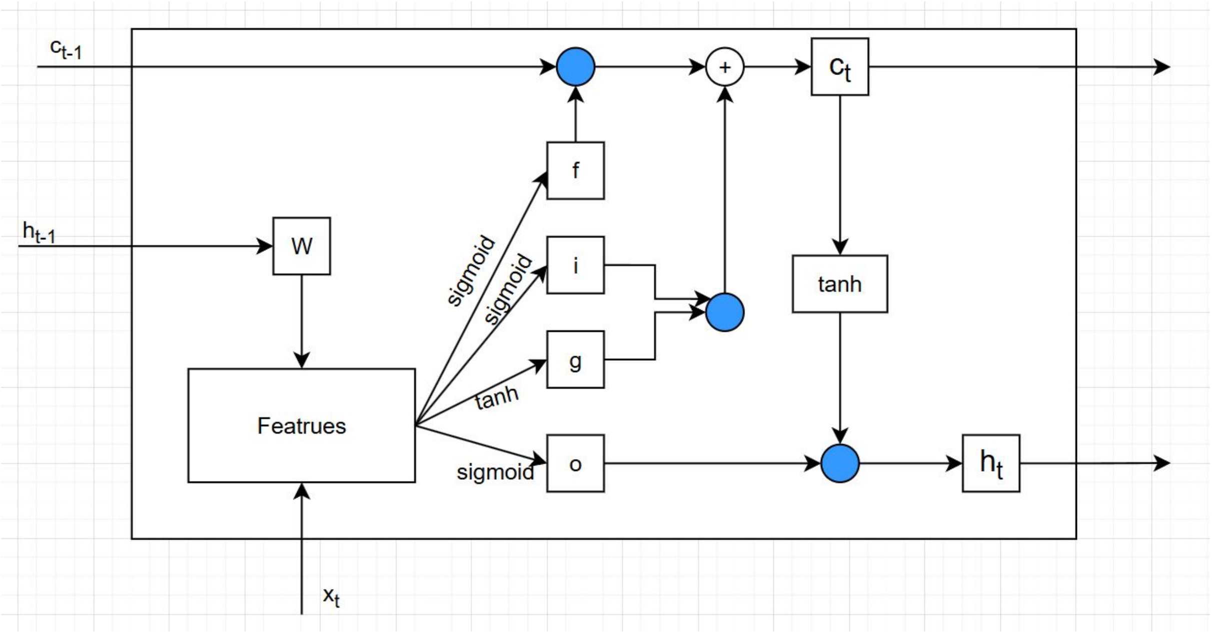
f = sigmoid(f)

g = tanh(g)

o = sigmoid(o)

$C_t = f \circ C_{t-1} + i \circ g$

h = o * tanh(c)



sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - LSTM Cell (1)

i = sigmoid(i)

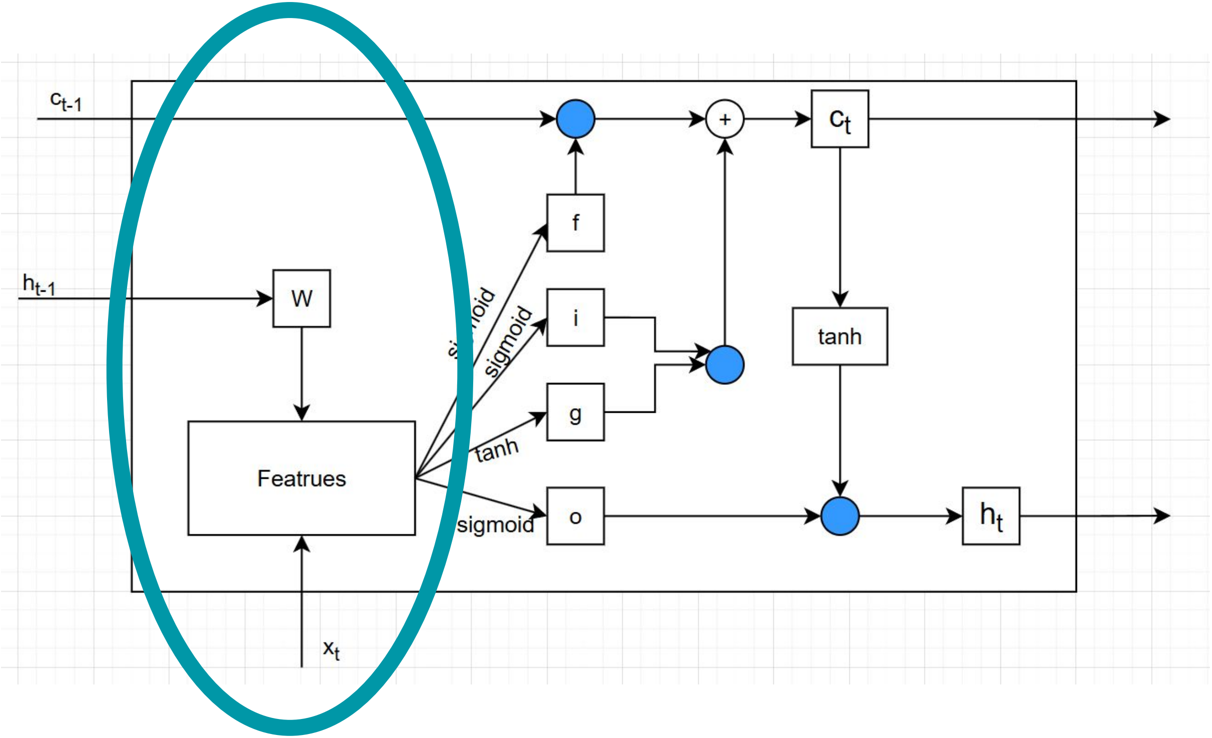
f = sigmoid(f)

g = tanh(g)

o = sigmoid(o)

$C_t = f \circ C_{t-1} + i \circ g$

h = o * tanh(c)



sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - LSTM Cell (2)

i = sigmoid(i)

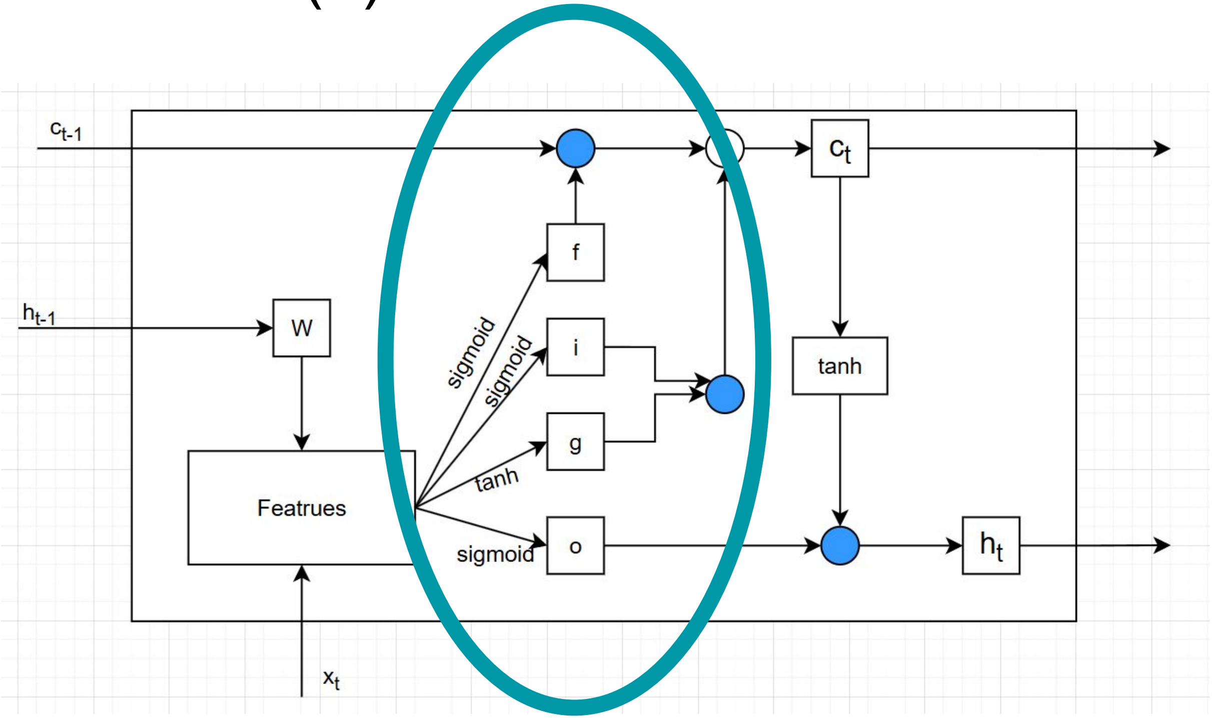
f = sigmoid(f)

g = tanh(g)

o = sigmoid(o)

$C_t = f \circ C_{t-1} + i \circ g$

h = o * tanh(c)



sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - LSTM Cell (3)

i = sigmoid(i)

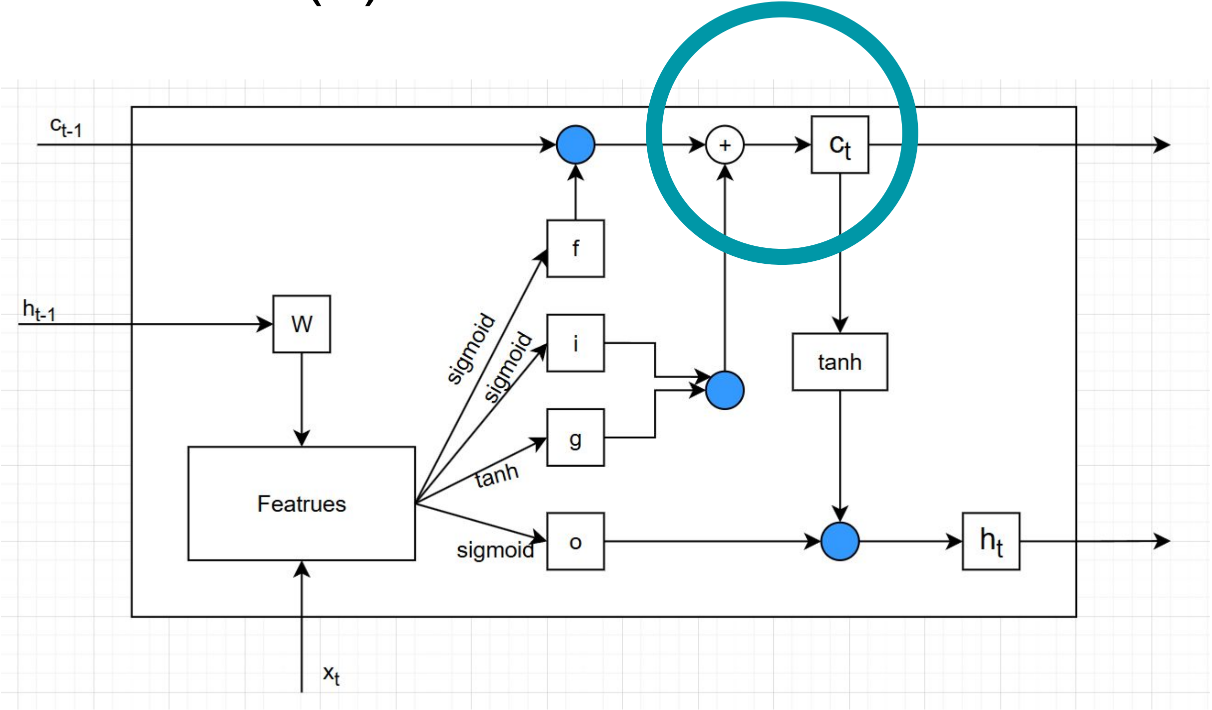
f = sigmoid(f)

g = tanh(g)

o = sigmoid(o)

$C_t = f \circ C_{t-1} + i \circ g$

h = o * tanh(c)



sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - LSTM Cell (4)

i = sigmoid(i)

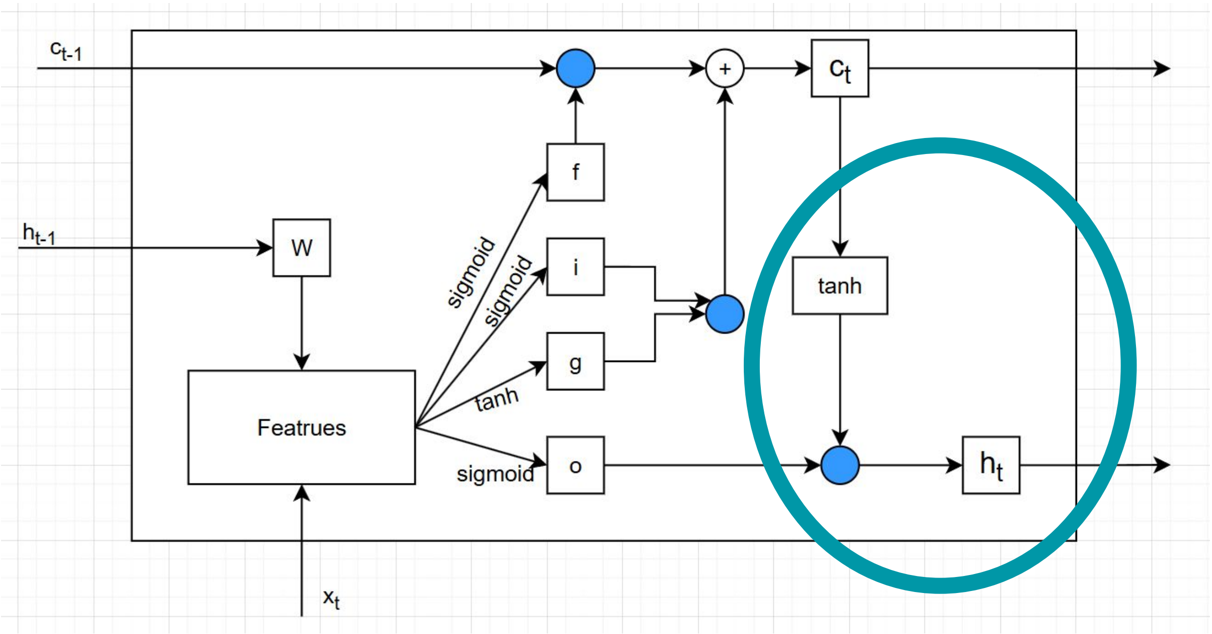
f = sigmoid(f)

g = tanh(g)

o = sigmoid(o)

$C_t = f \circ C_{t-1} + i \circ g$

h = o * tanh(c)



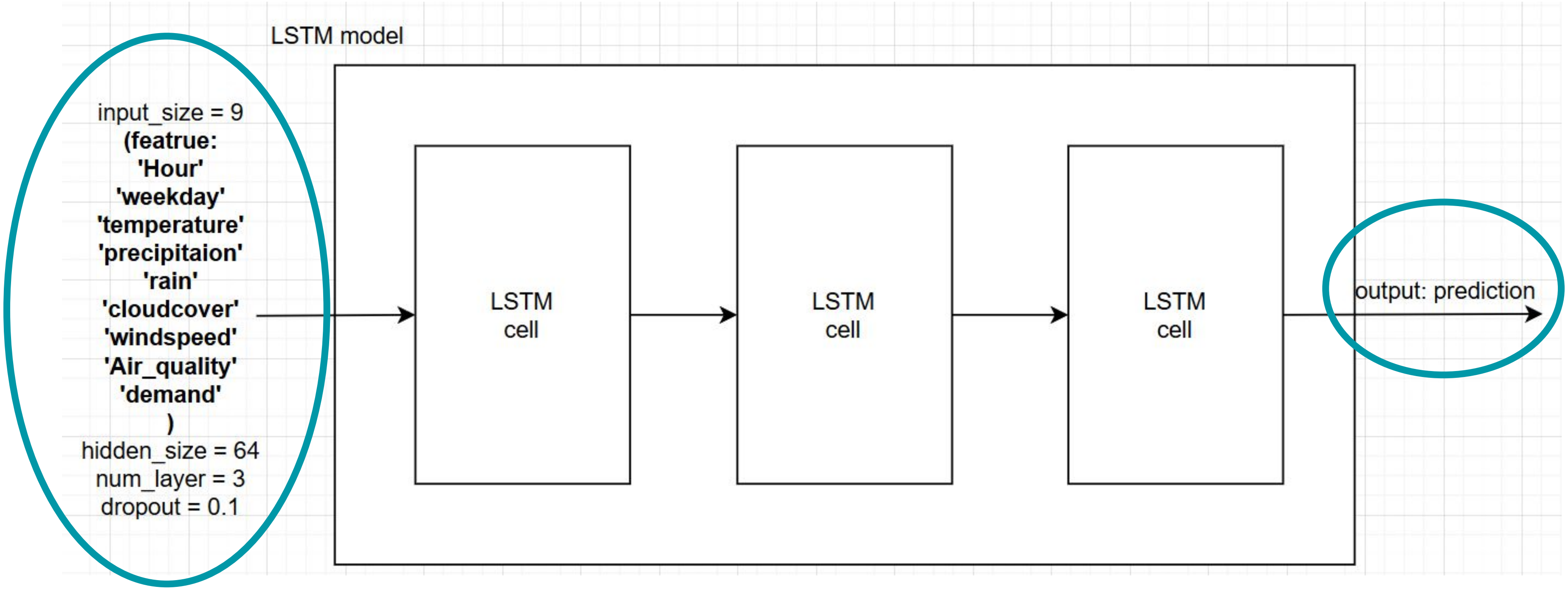
sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - LSTM Cell (5) - part of code

```
def forward(self, x, h_prev, c_prev):
    gates = self.x2h(x) + self.h2h(h_prev)                  #linear transformation
    i_gate, f_gate, g_gate, o_gate = gates.chunk(4, dim=1)
    i_gate = torch.sigmoid(i_gate)                  #input gate
    f_gate = torch.sigmoid(f_gate)                  #forget gate
    g_gate = torch.tanh(g_gate)                     #cell gate
    o_gate = torch.sigmoid(o_gate)                  #output gate
    c = f_gate * c_prev + i_gate * g_gate           #cell state : c = f ∘ c_prev + i ∘ g
    h = o_gate * torch.tanh(c)                      #hidden state
    return h, c                                     #pass to next cell
```

sigmoid : σ(x)=1 / (1+e^−x), tanh: tanh(x)=(e^x-e^−x)/(e^x+e^−x)

Main Approach - input & output



Main Approach - input & output

Input :

All : features = ['Hour', 'weekday', 'temperature', 'precipitation', 'rain', 'cloudcover', 'windspeed', 'Air_quality', 'demand']

Demand : features = ['Hour', 'weekday', 'demand']

Historical : features = ['Hour', 'weekday']

Weather : features = ['Hour', 'weekday', 'temperature', 'precipitation', 'rain', 'cloudcover', 'windspeed', 'Air_quality']

Output : our prediction, 3 evaluation metric (see in next page!).

Evaluation Metrics

3 type of evaluation metric are used to evaluate our models' performance and prediction accuracy

1. RMSE : measure the average magnitude of the differences (lower is better)
2. MAE : evaluate average difference (lower is better)
3. R^2 : measures the goodness of fit of our model to the data (higher is better)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

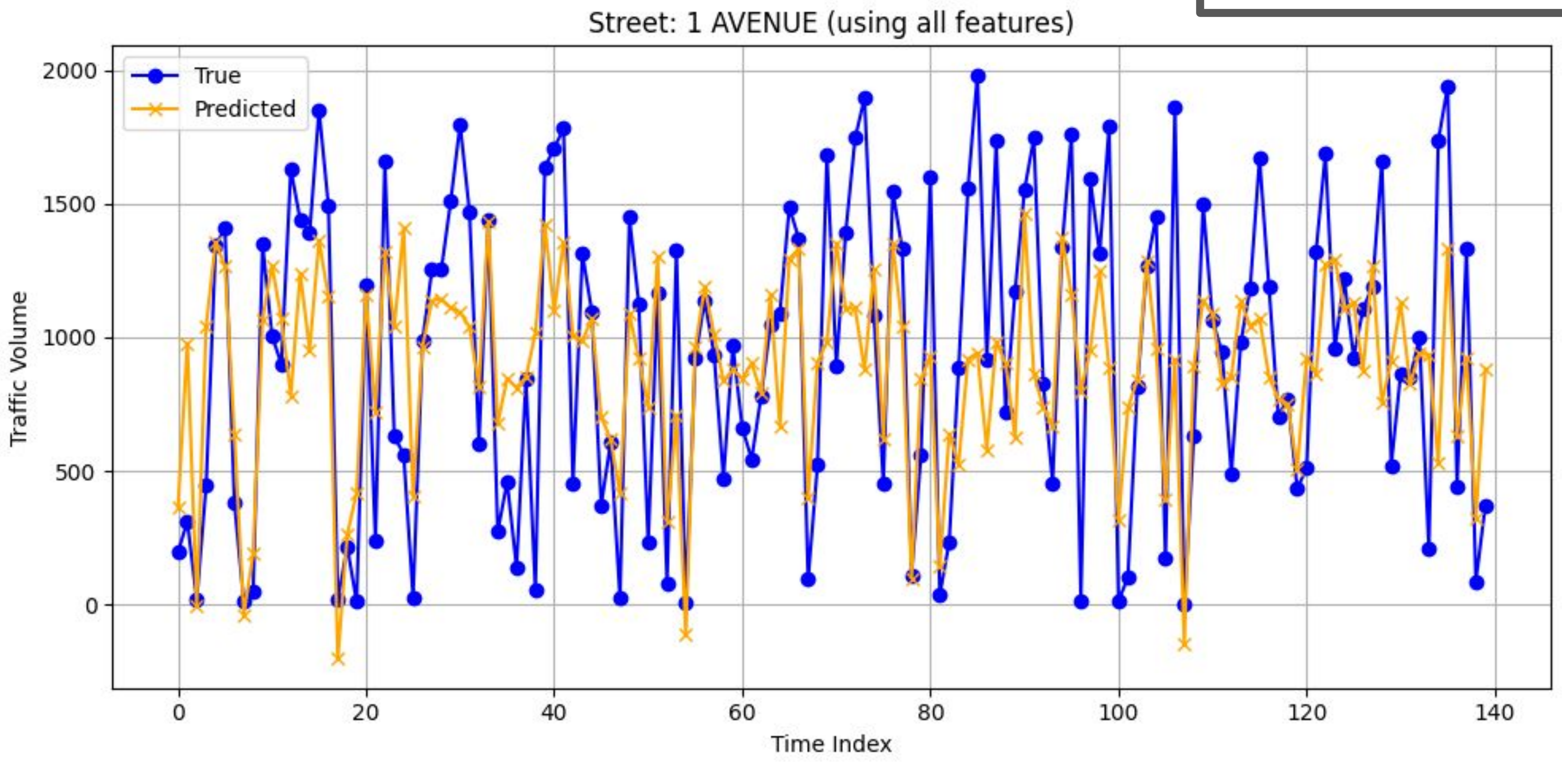
$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

RMSE is more sensetive to large error than MAE and R square

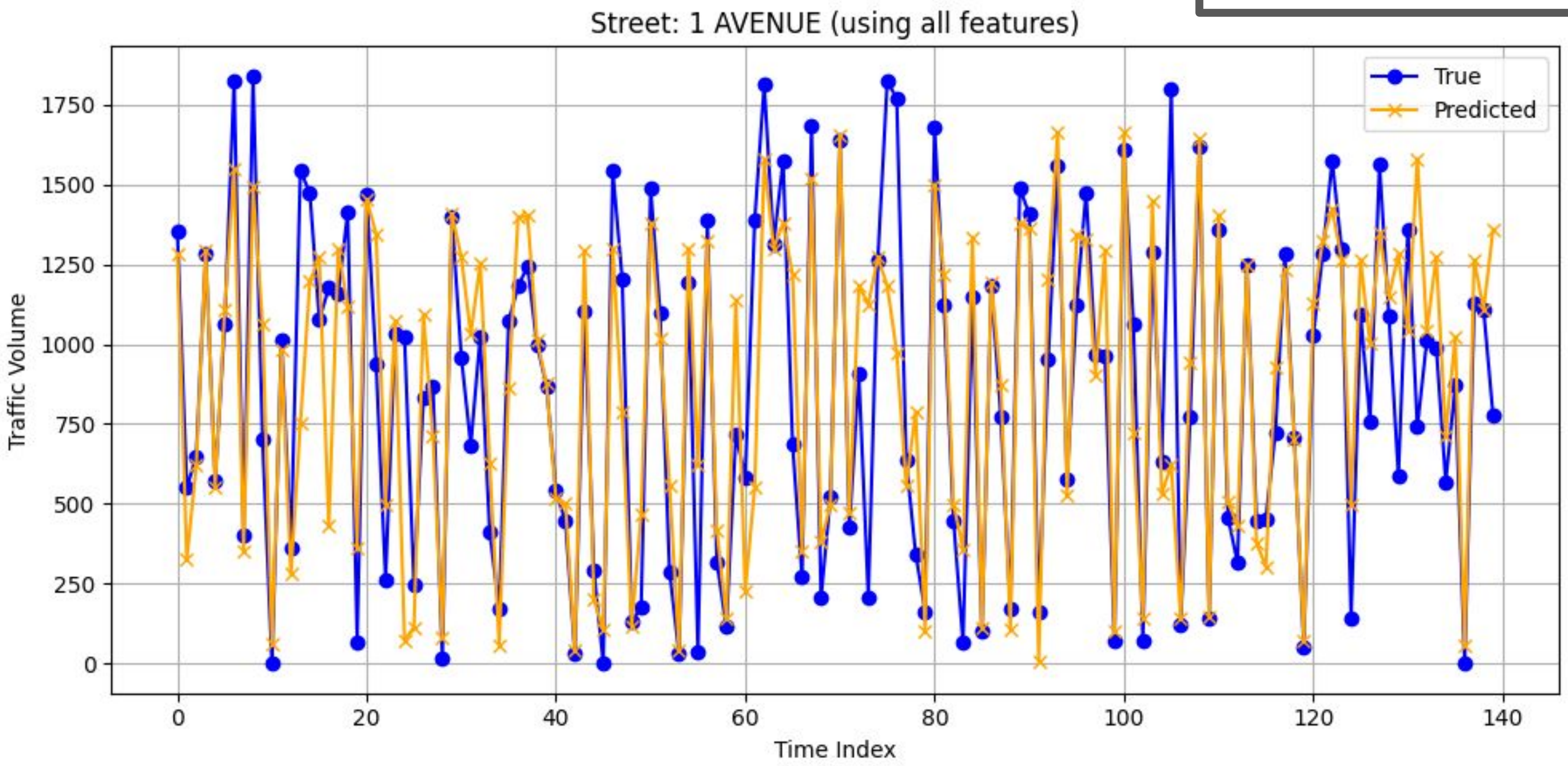
Results & Analysis - Baseline

RMSE: 344.7538
MAE : 217.9076
R square : 0.7011



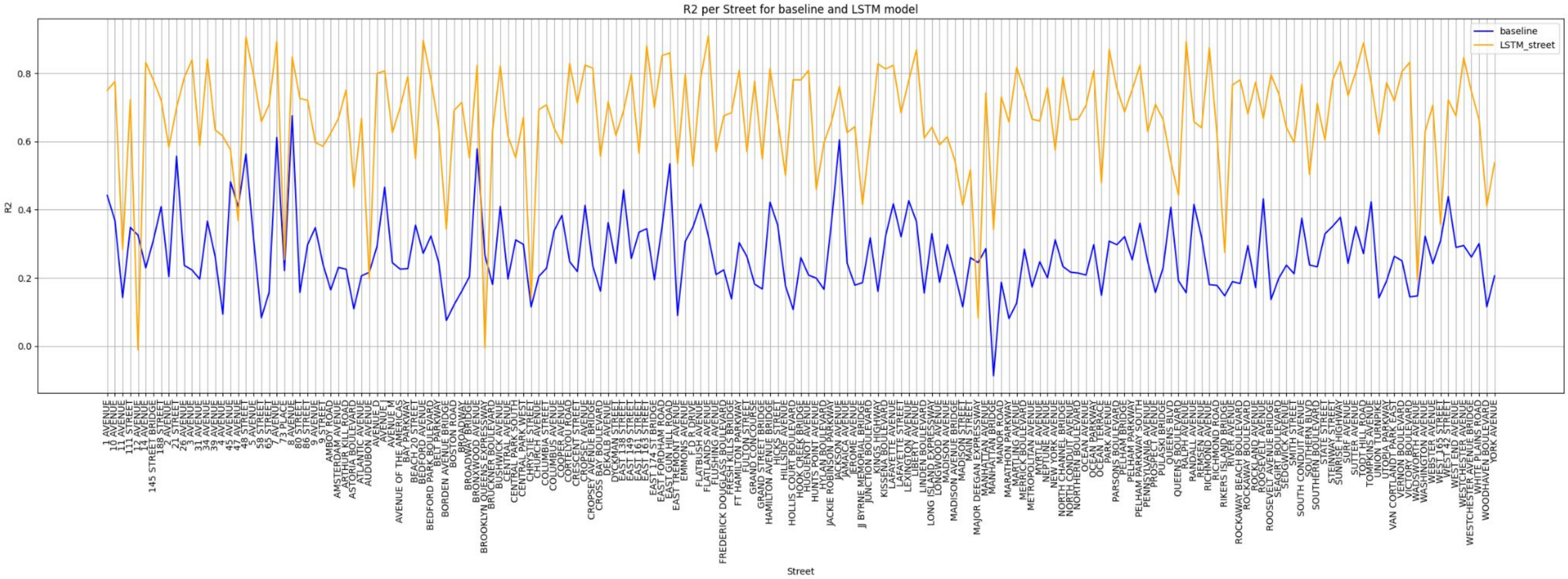
Results & Analysis - Main approach

RMSE: 241.9979
MAE : 129.6413
R square : 0.8527

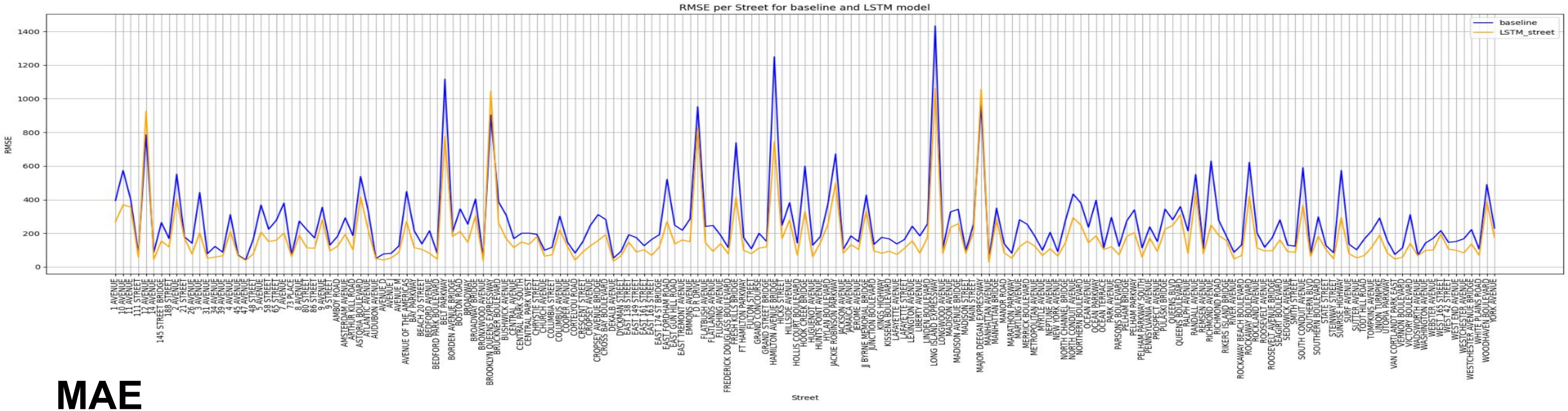


Results & Analysis - Comparison of evaluation metrics

R square

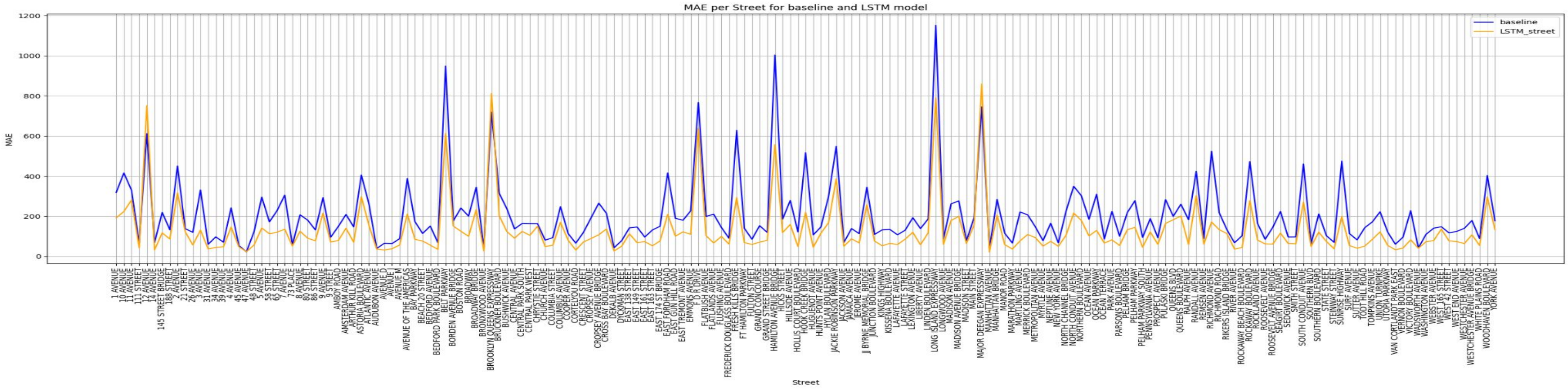


RMSE



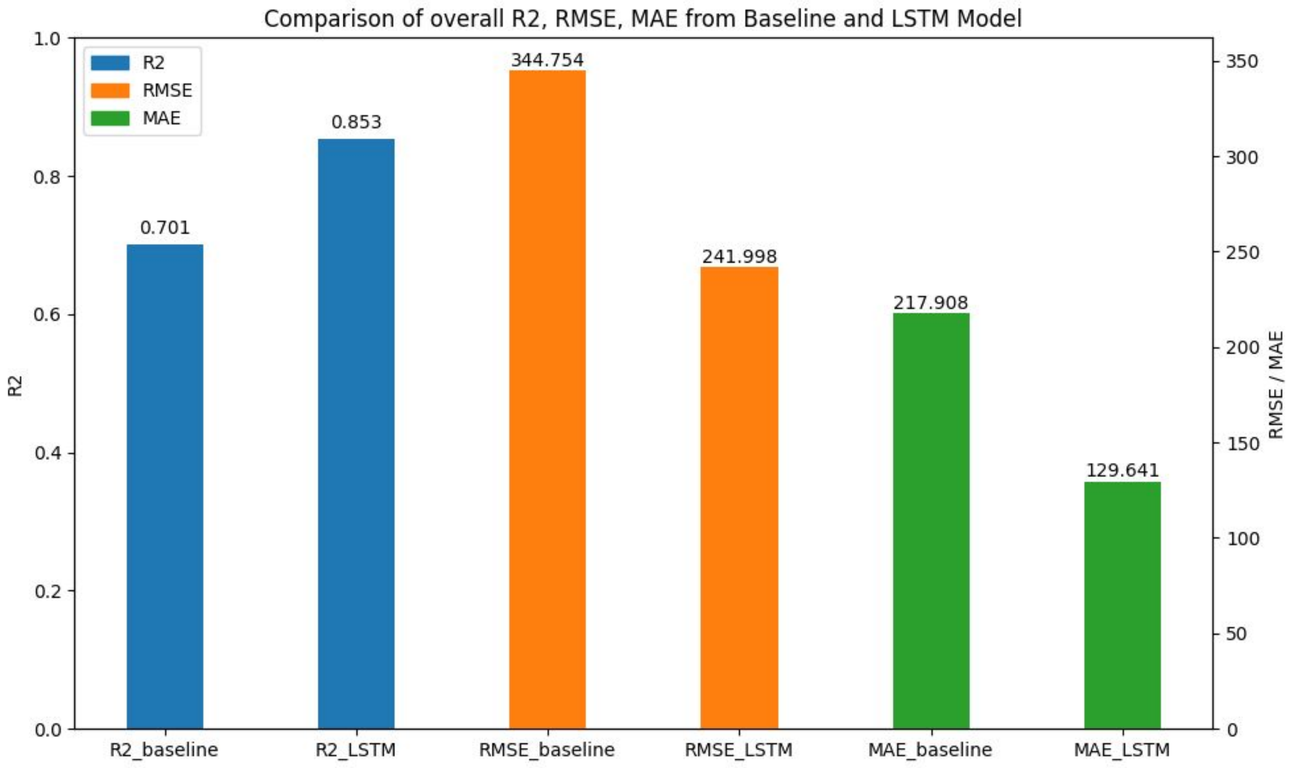
RMSE per Street for baseline and LSTM model
RMSE
Street
baseline
LSTM_street

MAE



MAE per Street for baseline and LSTM model
MAE
Street
baseline
LSTM_street

Results & Analysis - Overall comparison



Results & Analysis - Baseline vs. Main approach analysis

1. There is a significant difference between baseline and main approach by numerical data.

   We think that it strongly depends on some data is temporal dependency (weather, the volume on the specific street…)

2. We can observe that we use Linear regression model LSTM model and to compare their R square value has significant difference, but RMSE and MAE has the same trend but a little bit difference.

   We think that LSTM will be a better predictor but not much better than linear regression

3. There are still have some very interesting phenomenon : some peak in the graph linear regression outperform LSTM.

   We think that maybe the number of data of the street is very small or that street could have some linear relation to the feature. However, most of prediction in LSTM is better than baseline

Results & Analysis - Type of Experiment

In experiment part, we do some experiments in this section

1. We group some features in this experiment:
   baseline : use Linear regression and grouped by street
   group : use LSTM architecture and grouped by Boro + street
   street : use LSTM architecture and grouped by street
   global : use LSTM architecture **but without grouped**

2. We adjust some parameter to observe the outcome of our experiment by comparing evaluation metrics.

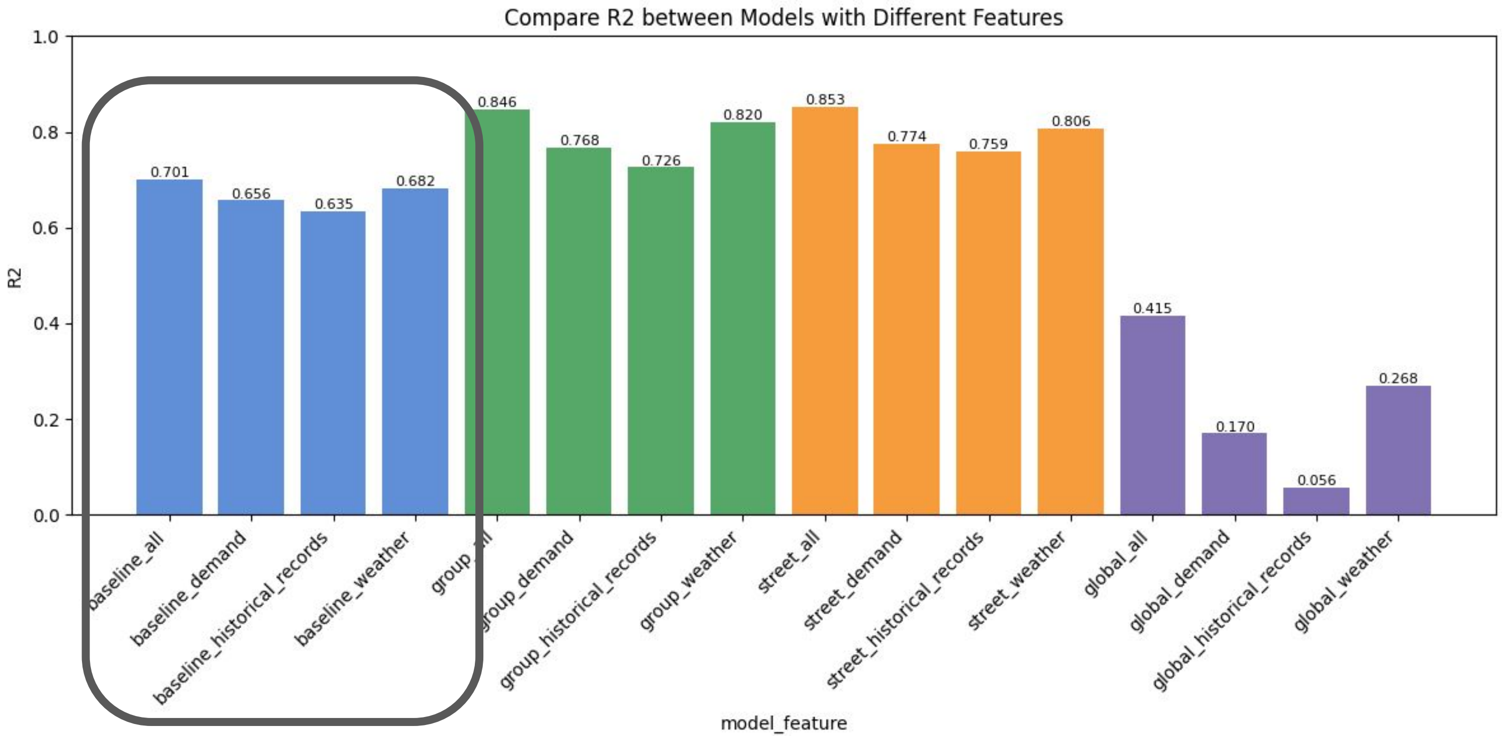
Results & Analysis - Experiment 1

In experiment type 1, we aim to **find the relationship between those factors and traffic volume and using different grouping method**.

We built several models for different grouping method that takes different variables as featrue to train and test, then plot its RMSE, MAE, R2 to show the difference between models under different situation.

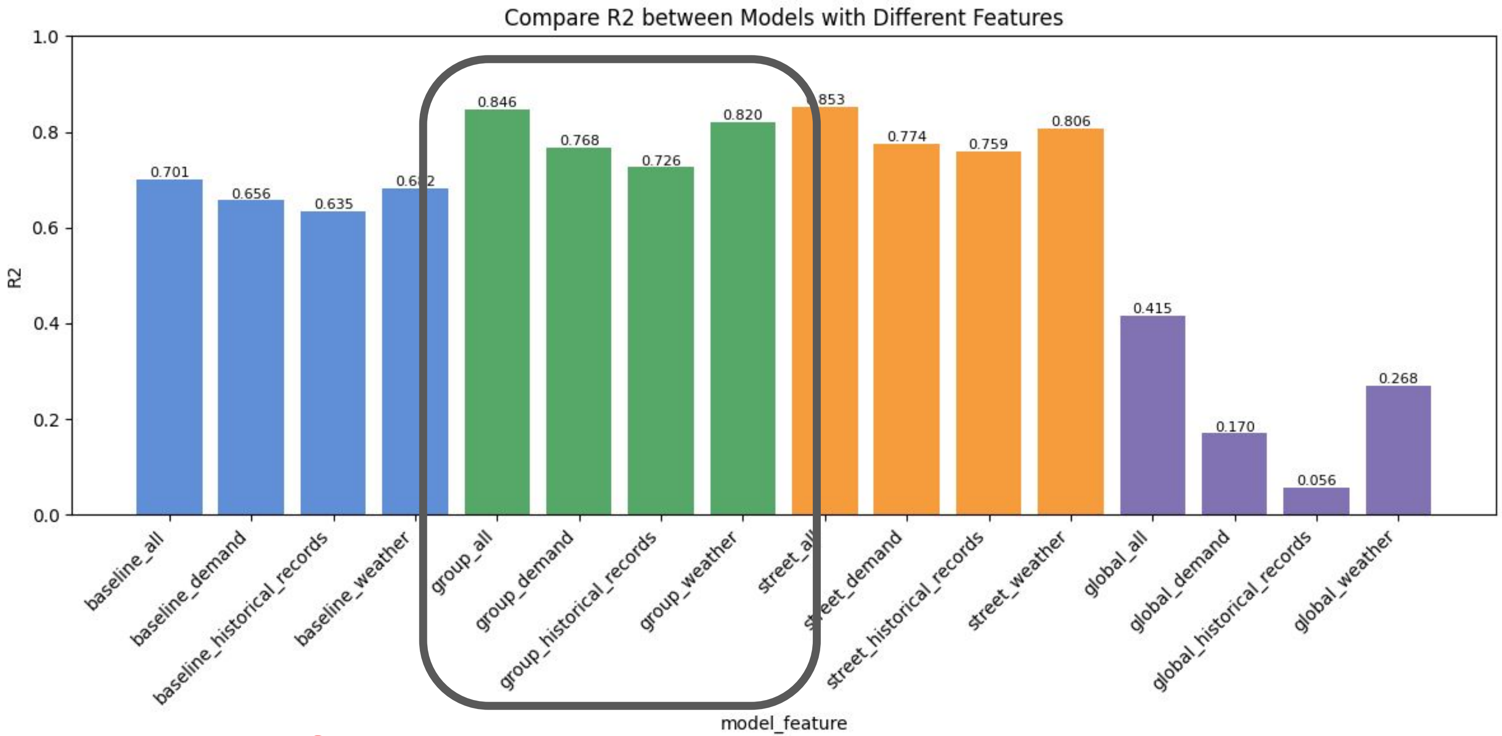
The following figure shows that the results of training using different grouping method and using different kinds of features combinations

Results & Analysis - Experiment1(R square) - baseline



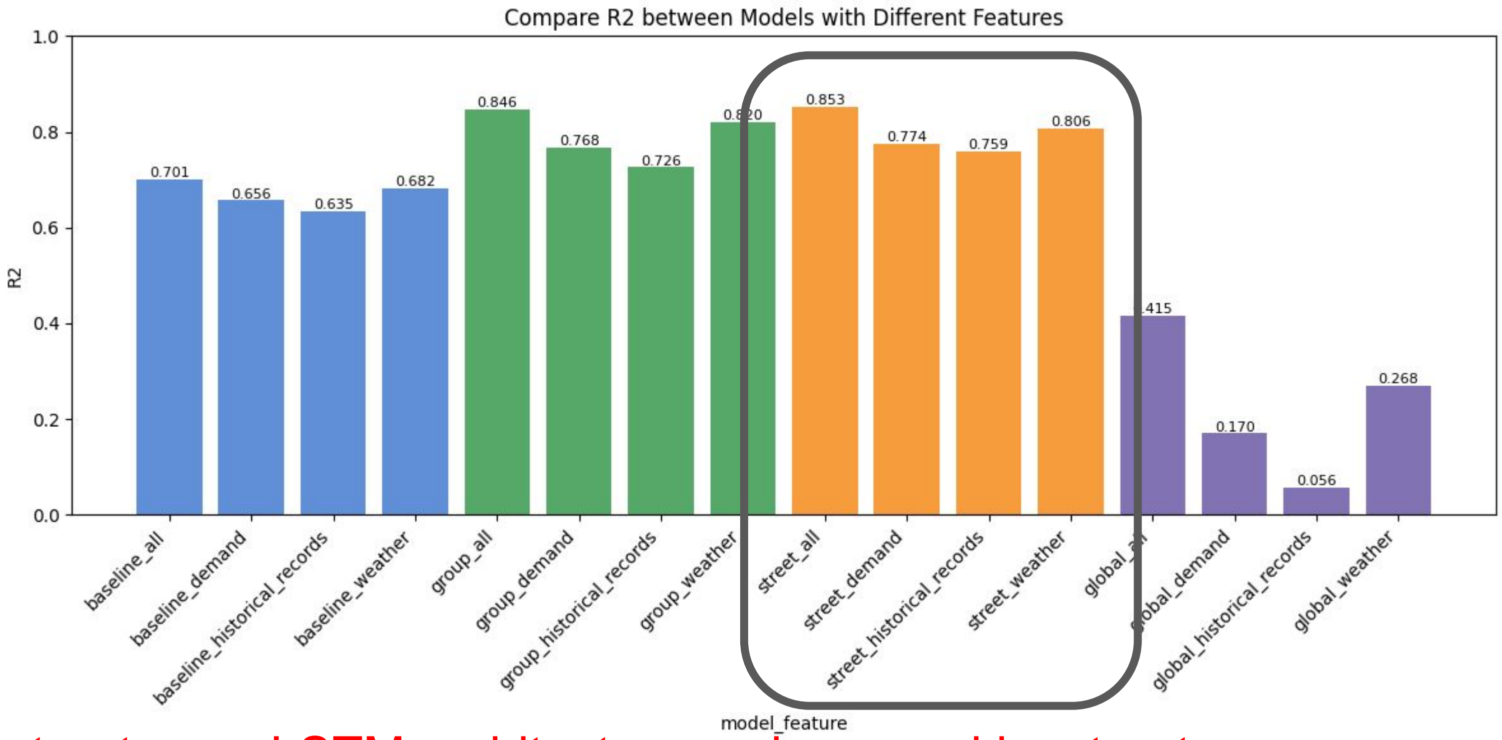
baseline : use Linear regression and grouped by street

Results & Analysis - Experiment1(R square) - group



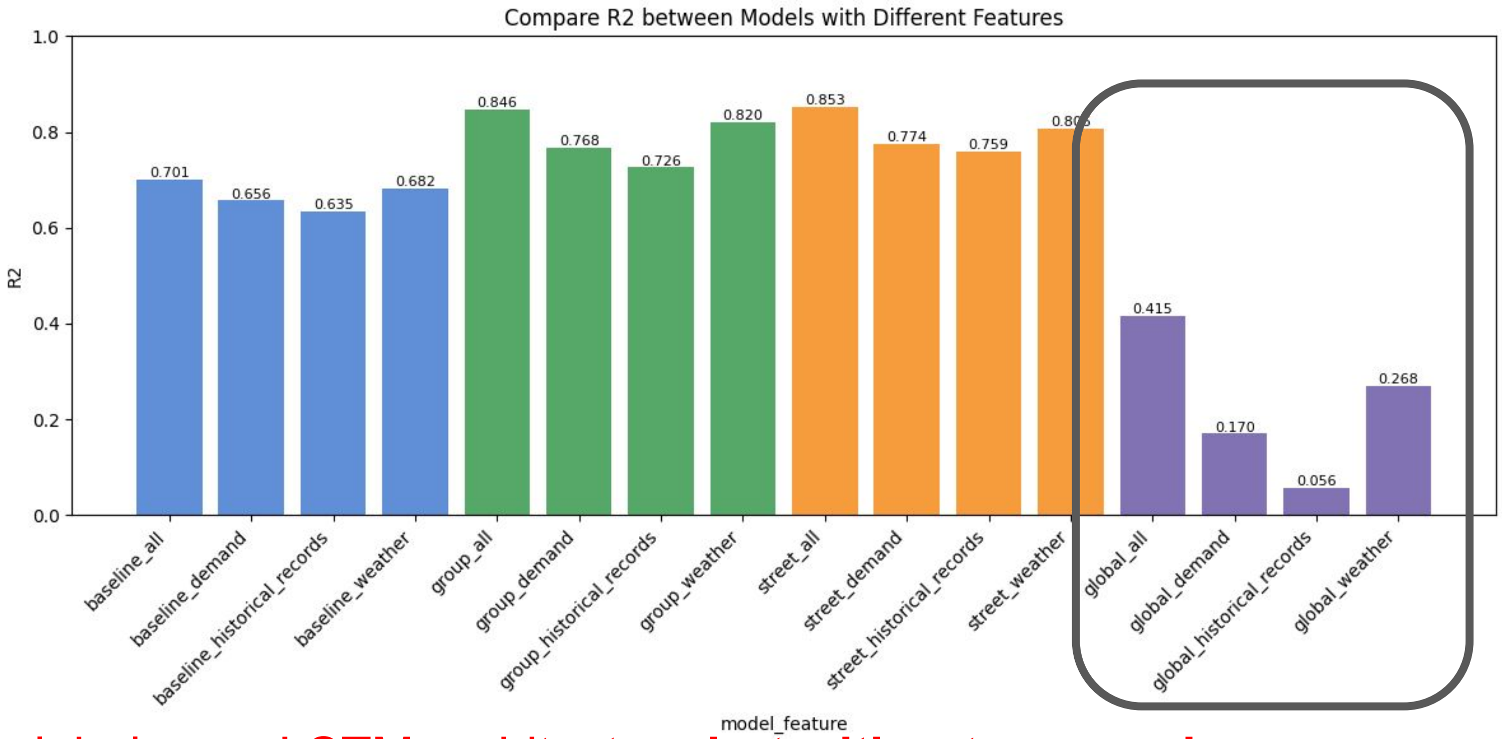
group : use LSTM architecture and grouped by Boro + street

Results & Analysis - Experiment1(R square) - street



street : use LSTM architecture and grouped by street

Results & Analysis - Experiment1(R square) - global



global : use LSTM architecture **but without grouped**

Results & Analysis - Experiment 1 result (R2) analysis

1. We can observe that the R2 of global is much less than the others R2.

   We think that because it do not be grouped, its predictions to all streets mix together and finally become a worst one in this experiment comparing to another data which is grouped by other features.

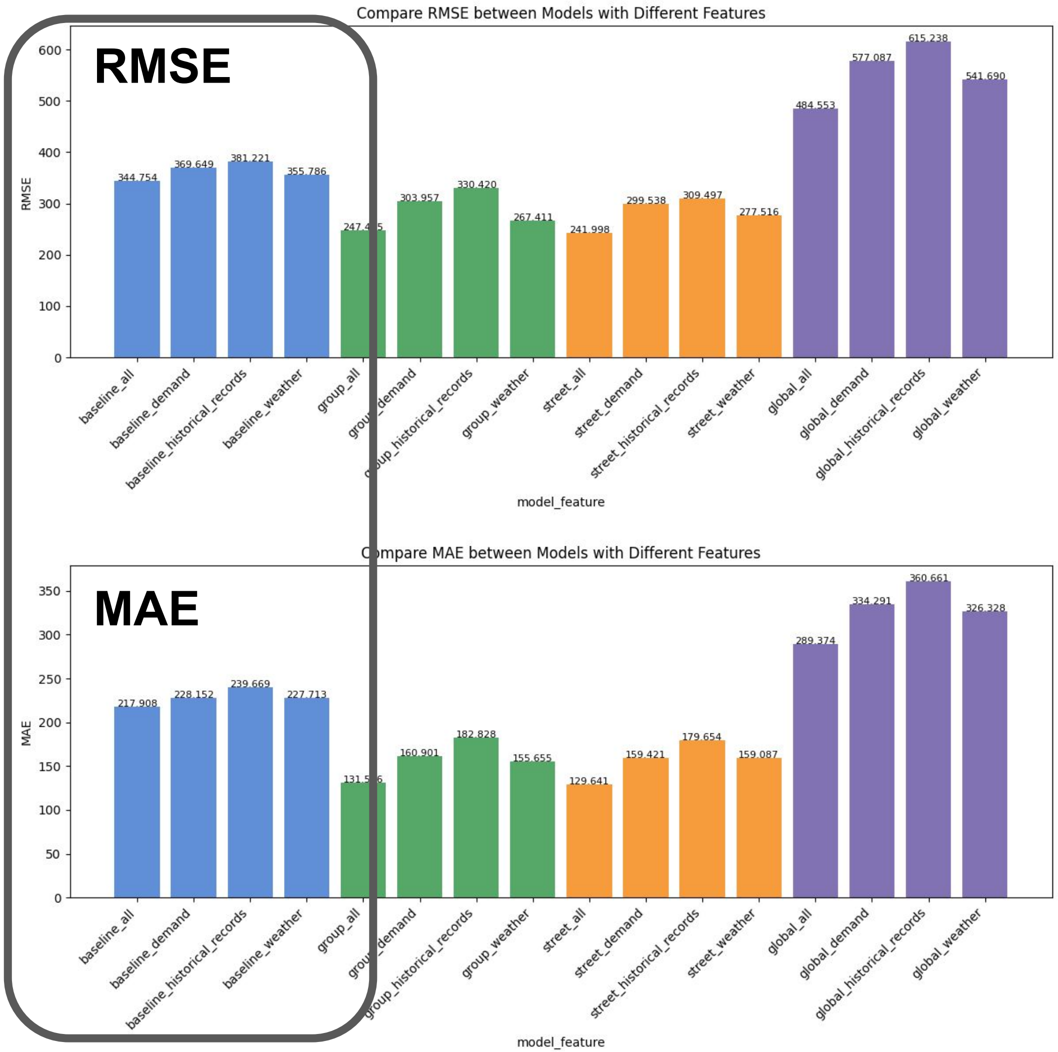
2. The group and street prediction slightly outperform the baseline prediction and the trend of them is very similar.

   We think that it proves our prediction and prove that traffic can be predicted in our model properly. They have similar trend because they use same features to train the model.

3. The R2 value of group and street is similar in the same input(feature).

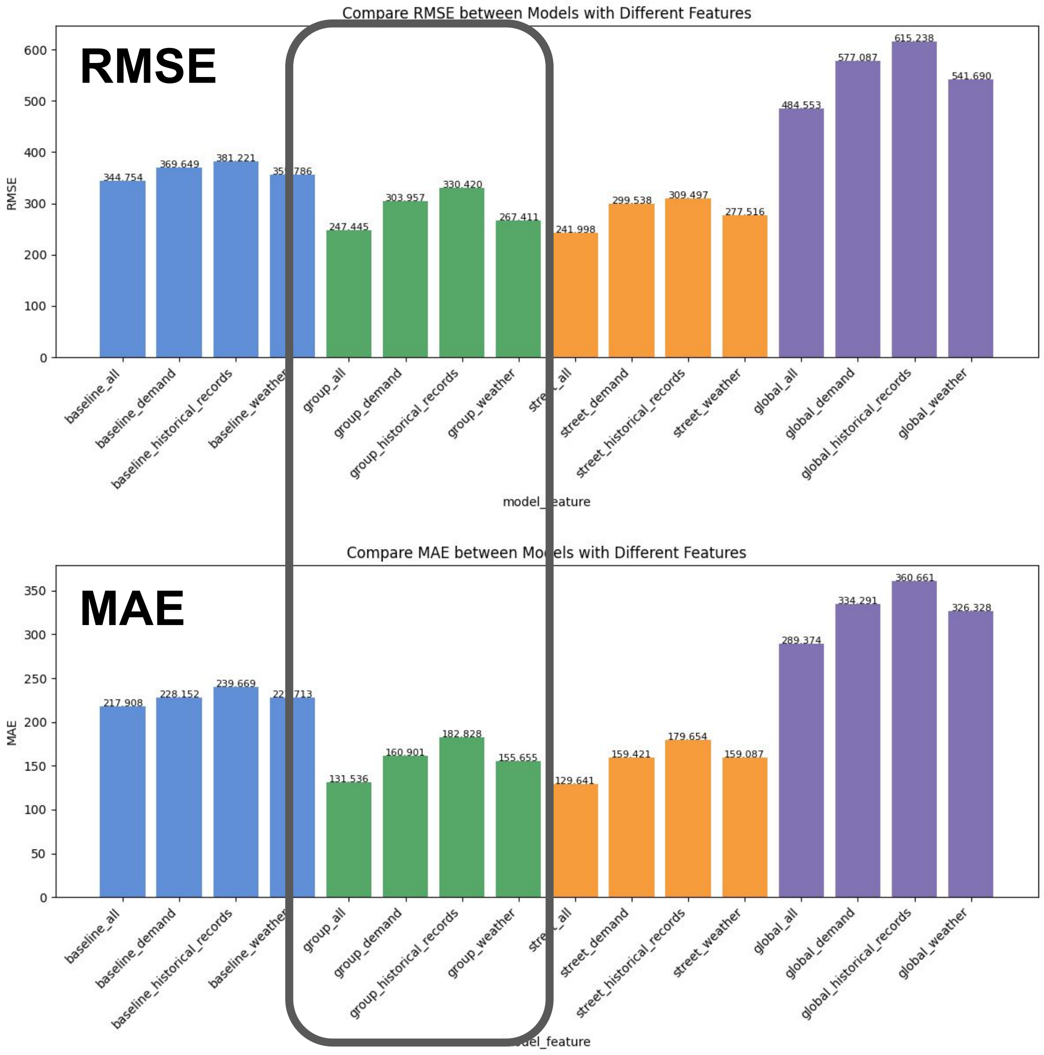
   We think that maybe “street” is the key point or “Boro” is independent to train this model. Another reason is that street has more dataset to train while group is not.

Results & Analysis - Experiment 1 result (RMSE, MAE) - baseline



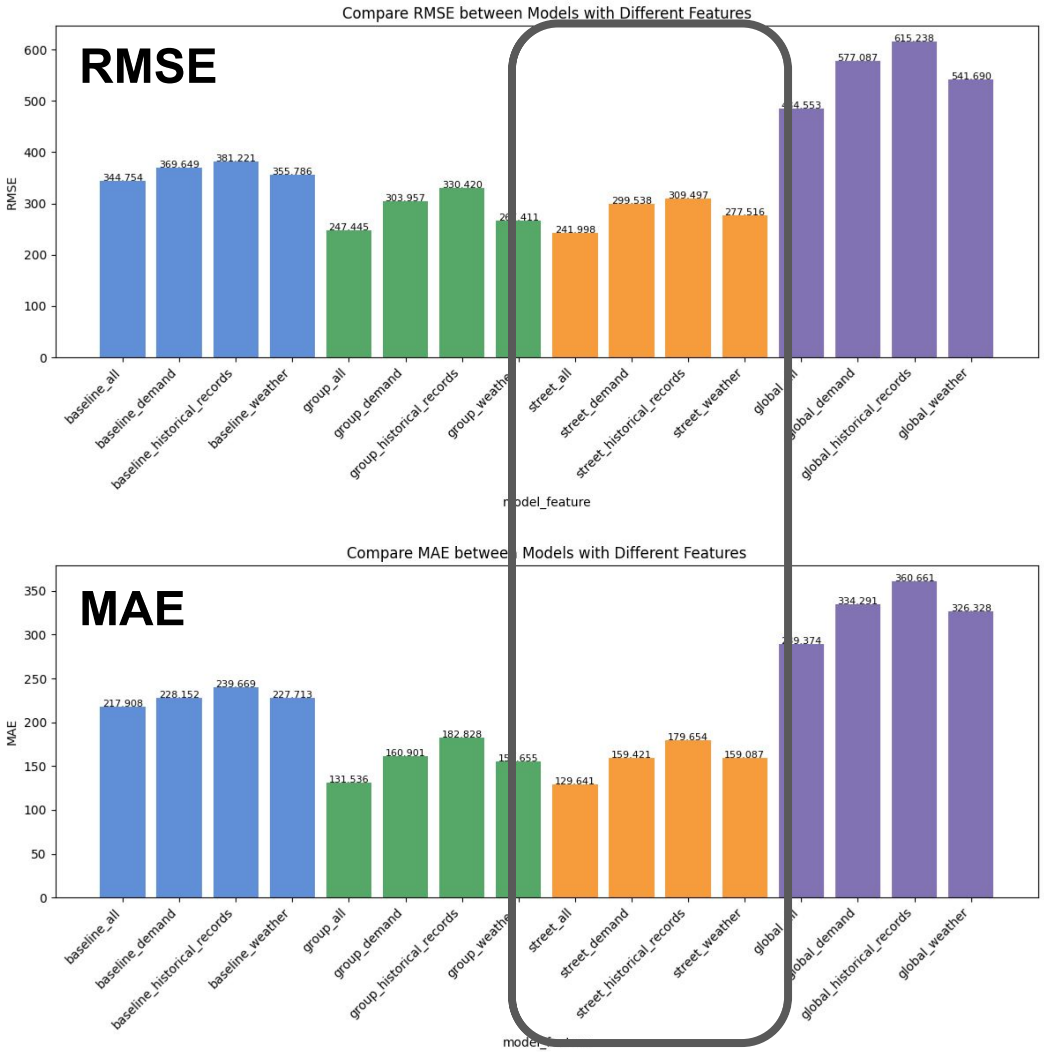
baseline : use Linear regression and grouped by street

Results & Analysis - Experiment 1 result (RMSE, MAE) - group



group : use LSTM architecture and grouped by Boro + street

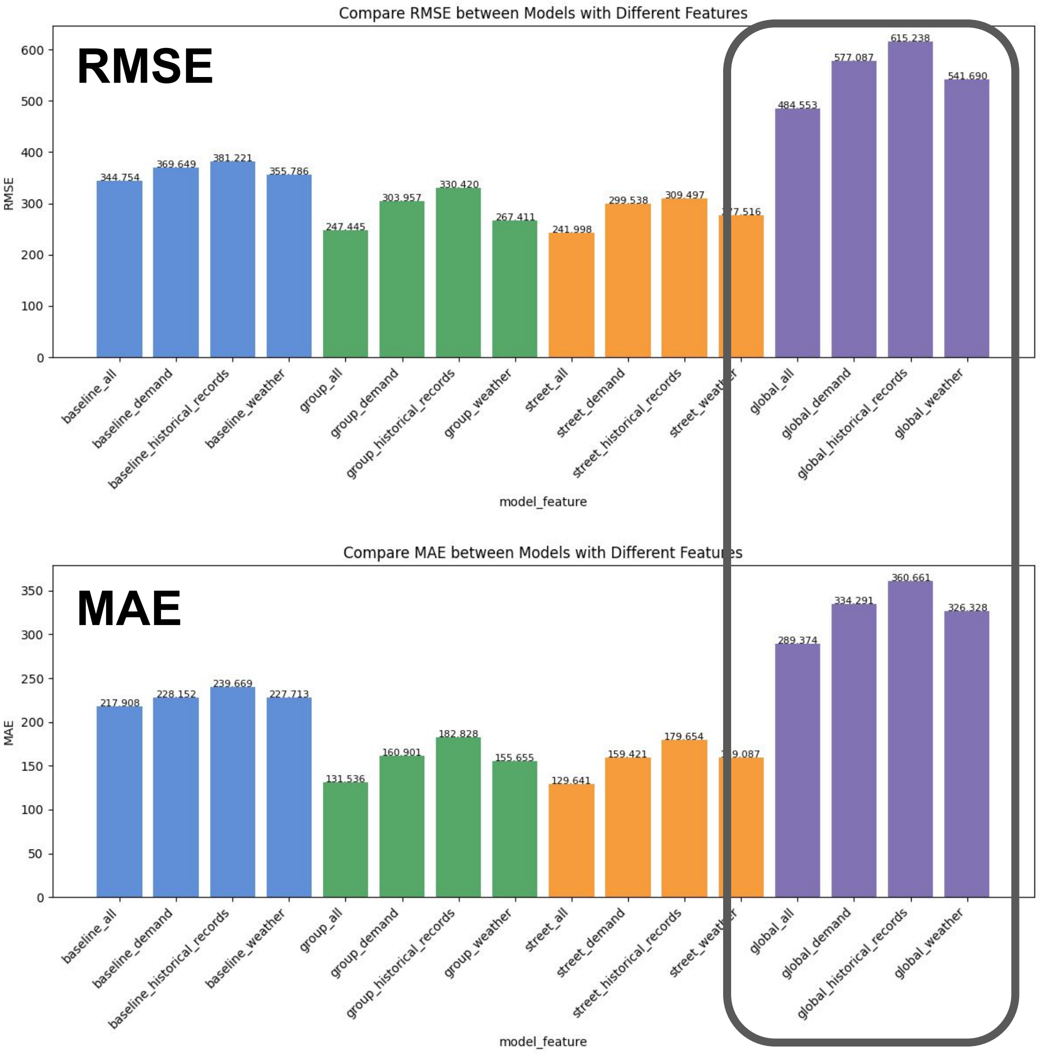
Results & Analysis - Experiment 1 result (RMSE, MAE) - street



street : use LSTM architecture and grouped by street

Results & Analysis - Experiment 1 result (RMSE, MAE) - global

global : use LSTM architecture **but without grouped**



Results & Analysis - Experiment 1 (RMSE, MAE) result analysis

1. All RSME and MAE on prediction in their feature is very similar to each other.

   We think that it because it does not have a lot of big error(high variance outliers) to make RSME penalize the training process, it may be small and consistent. Another reason is that the model could be uniformly underfit.

2. The global RSME and MAE is much higher than the others' RSME and MAE

   We think that because it do not be grouped, its predictions to all streets mix together and finally become a worst one in this experiment comparing to another data which is grouped by other features.

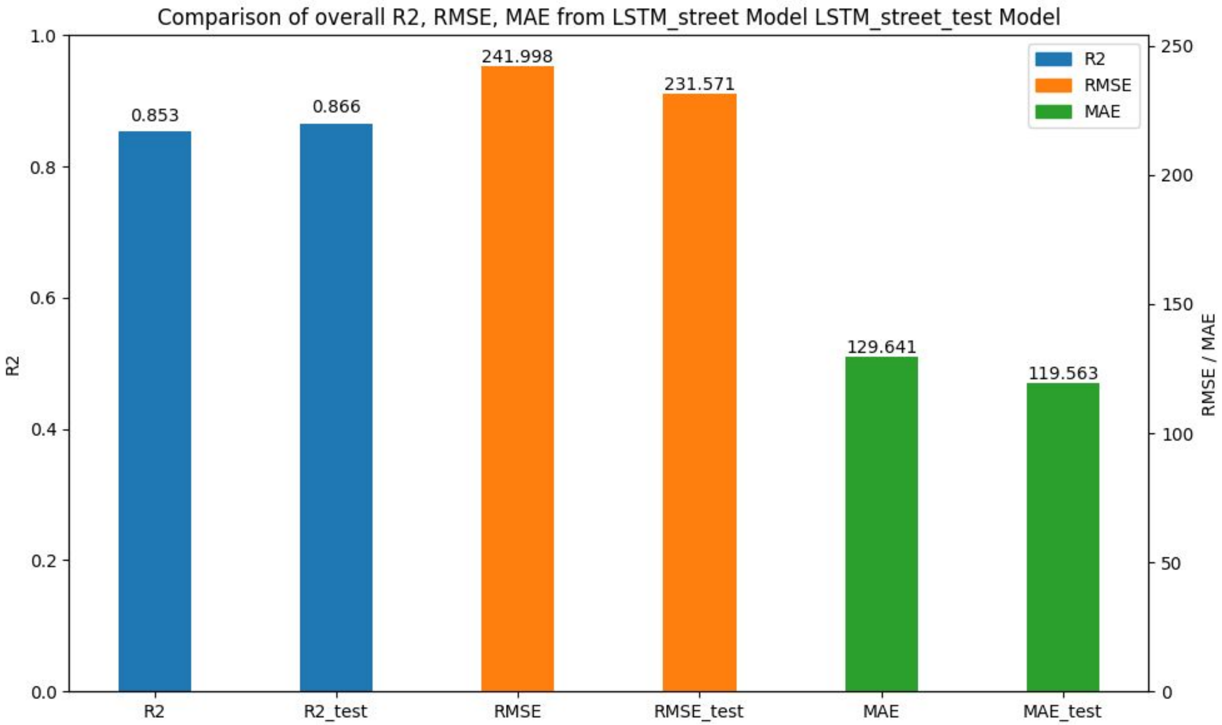
3. The trend in RMSE is very similar to each group (so as MAE) and the group and street prediction slightly outperform the baseline prediction

   We think that it proves our prediction and prove that traffic can be predicted in our model properly. They have similar trend because they use same features to train the model.

Results & Analysis - Experiment 2

We also adjust some parameters in this experiment.

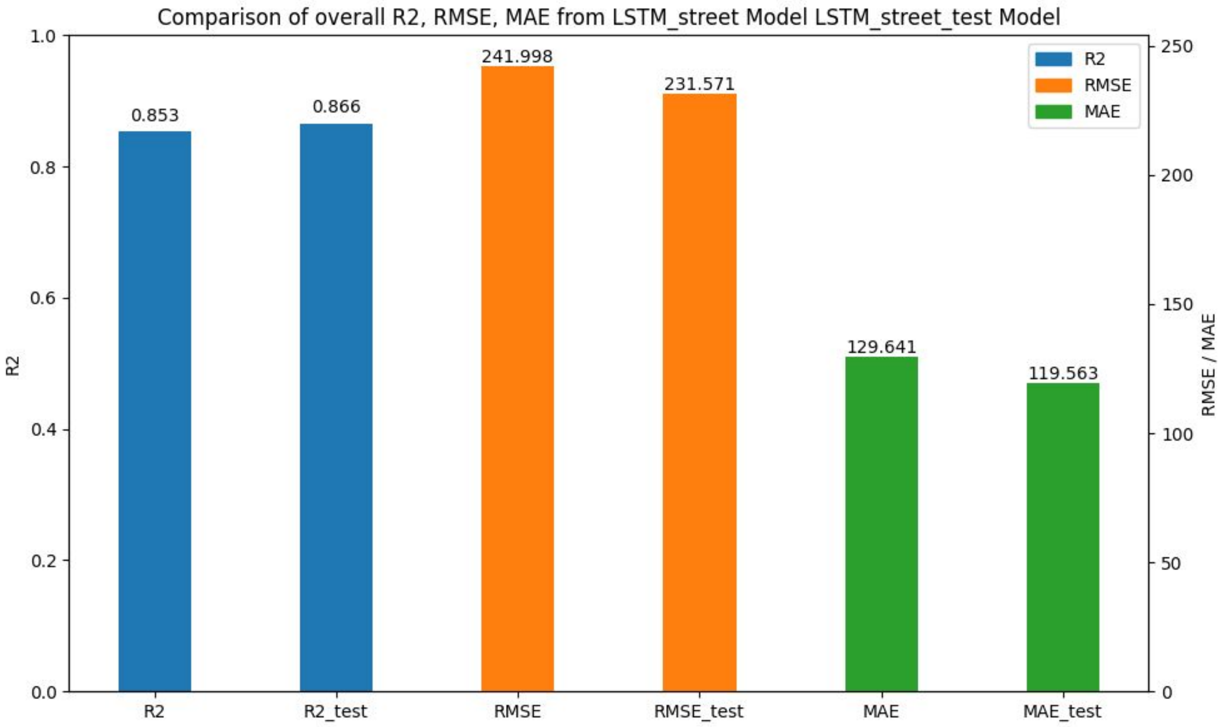
1. **increase the epoch (20 to 30)**
2. **increase number of cell layers (3 to 4)**
3. **add dropout layer in fully-connected with rate = 0.1.**



Results & Analysis - Experiment 2

Analysis:

We observe that R2 slightly increase, while RMSE and MAE slightly decrease because it trains much time and learn more to try to fit it.



Results & Analysis - Limitation of our work

1. Our datasets might not be large enough because we only have data from 2016 to 2022.
2. Because the data in NYC is much more than other cities, we take NYC for example. However, it bring some limitation on this final project due to its specificity in NYC.
3. The time scale of our prediction is one prediction per hour due to our dataset, so our project cannot predict one time in a closer range, for instance, 30 minutes.

Results & Analysis - The real world application

We use API(Application Programming Interface) to capture the real time information that will be used in prediction of traffic.

Get some data from Google weather API, such as temperature, precipitation, rain, cloudcover, and windspeed. And get the Air quality from IQair. Thus, we can achieve the real world application to predict the traffic for the next 6 hours.

We still use our code (grouped by street use all of the features to train model).

Our API Sources :
https://developers.google.com/maps/documentation/weather/overview?hl=zh-tw
https://api-docs.iqair.com/?version=latest

Results & Analysis - Apply the model in the real world

- Using terminal to show the prediction
- Eastern Daylight Time (The time zone of NYC)

```
weichen@LAPTOP-655HRAQS:~/AI/project/AI_Group11_Final_Project$ python3 application/api_application.py
Enter the street name: 1 avenue

Predicted traffic for street '1 AVENUE' for the next 6 hours:
Hour 06:00 - Predicted volumn: 557.14
Hour 07:00 - Predicted volumn: 570.67
Hour 08:00 - Predicted volumn: 584.62
Hour 09:00 - Predicted volumn: 598.54
Hour 10:00 - Predicted volumn: 614.25
Hour 11:00 - Predicted volumn: 630.26
```

GitHub Link

https://github.com/k77914/AI_Group11_Final_Project

References

- LSTM concepts: https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/
- LSTM reference scratch: https://medium.com/@wangdk93/lstm-from-scratch-c8b4baf06a8b
- AirVisual API: https://api-docs.iqair.com/?version=latest
- Google Weather API: https://developers.google.com/maps/documentation/weather/overview?hl=zh-tw

Contribution - what we mainly do & proportion distribution

蔡烝旭: Real world application, ReadMe, Github management (proportion : 22%)

徐瑋晨: Dataset, Main approach, experiments (proportion : 34%)

曾歆喬: Baseline model, Plot (proportion : 22%)

劉彥廷: Presentation, Powerpoint slide (proportion : 22%)

Thanks for your attention!