EAT for Bus

Using Machine Learning Methods

Agenda

Topics Covered

- Introduction
- Data we have
- Data Preprocessing
- Exploratory Data Analysis
- Models & results
- Conclusion
- Future work

Introduction

Objective

Developing and evaluating machine learning models for accurately predicting bus arrival time at traffic light stop lines.

Key Goals:

- Data Preprocessing
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Model Development
- Model Evaluation

Fig 1:Bus at Traffic Lights

Background

1. Schedule-Based Predictions [1]

- Buses are expected to adhere to a timetable
- Advantages: Easy to implement and understand
- Limitations: Fails to account for real-time variations

2. Historical Data Analysis [2]

- Utilizes historical travel time data to predict future arrivals, it averages travel times over similar conditions
- Advantages: More reliable than schedule-based predictions as it incorporates real travel time data.
- Limitations: Historical averages can smooth out significant variations and fail to adapt to real-time changes

3. Bus Priority Signal System [3]

more advanced traditional method aimed at improving the punctuality of bus services

- Bus Detection and Priority Request
- Traffic Signal Adjustment
- Restoration of Normal Timing

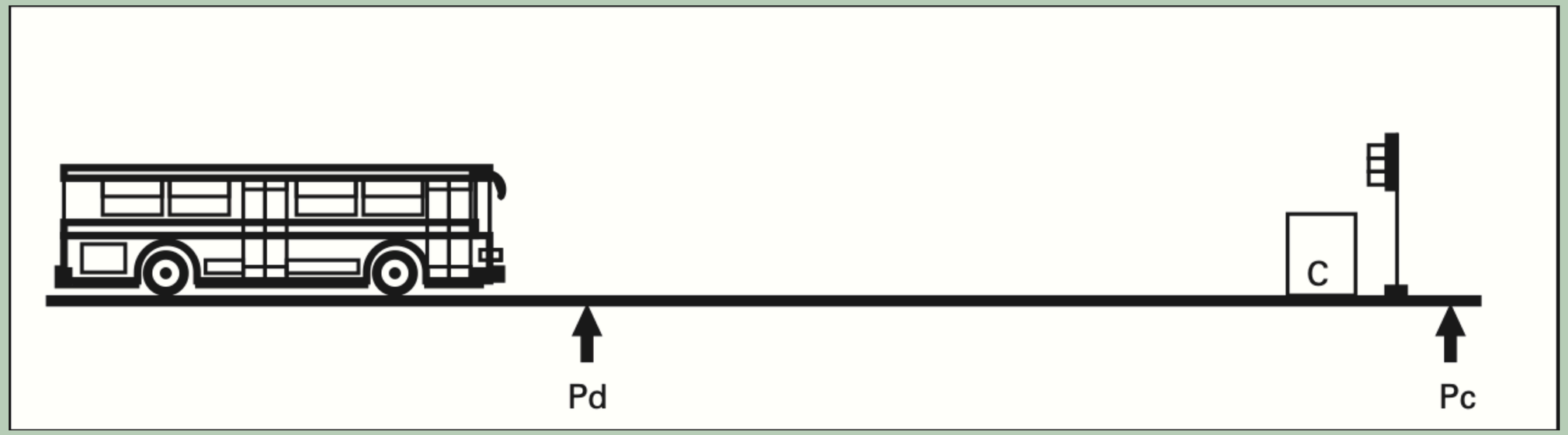


Fig 2:Bus Priority Signal System [3]

Related work

Bus Arrival Time Prediction Using Linear Regression [4]

- Dataset: a GPS dataset for buses from Dublin.
- features: ’day of the week’, ’hour’ and calculated distance from the city center.
- For model training, they used 80% and 20% for testing.
- Evaluation: MAE, MAPE, and RMSE.
- MAE: 7.6130 minutes, MAPE: 7.2406%, and RMSE: 10.0342 minutes

Bus Arrival Prediction Using Artificial Neural Network [5]

- Structure:
  -Input layer
  -One hidden layer (5 neurons)
  -Sigmoid activation function
- Data used:
  -Macao, China
  -Bus location
  -Weather conditions
  -Duration between bus stops
- Results:
  -MAPE 17%
  -MAE & RMSE less than a minute

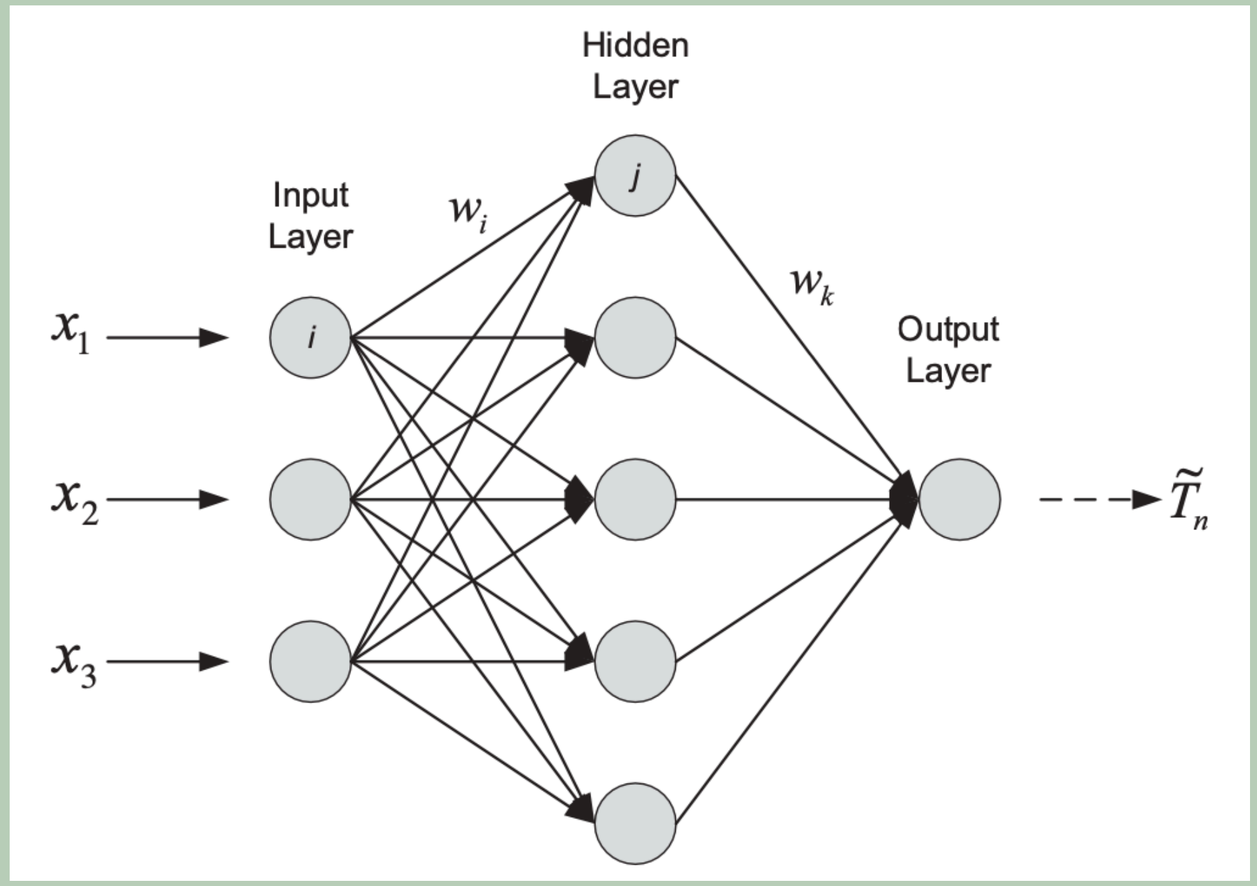


Fig 3:Model structure [5]

Bus Arrival Prediction Using LSTM [6]

- Structure:Input layer, two hidden layers (50 cells each) with tanh activation function, Adam optimizer and dropout layers to prevent overfitting.
- Data used: Chongqing, China. The model included features such as bus ID, speed, timestamp, longitude, latitude, dwell time at bus stops and traffic conditions.
- Results:

-MAPE around 7%

-RMSE around 40s

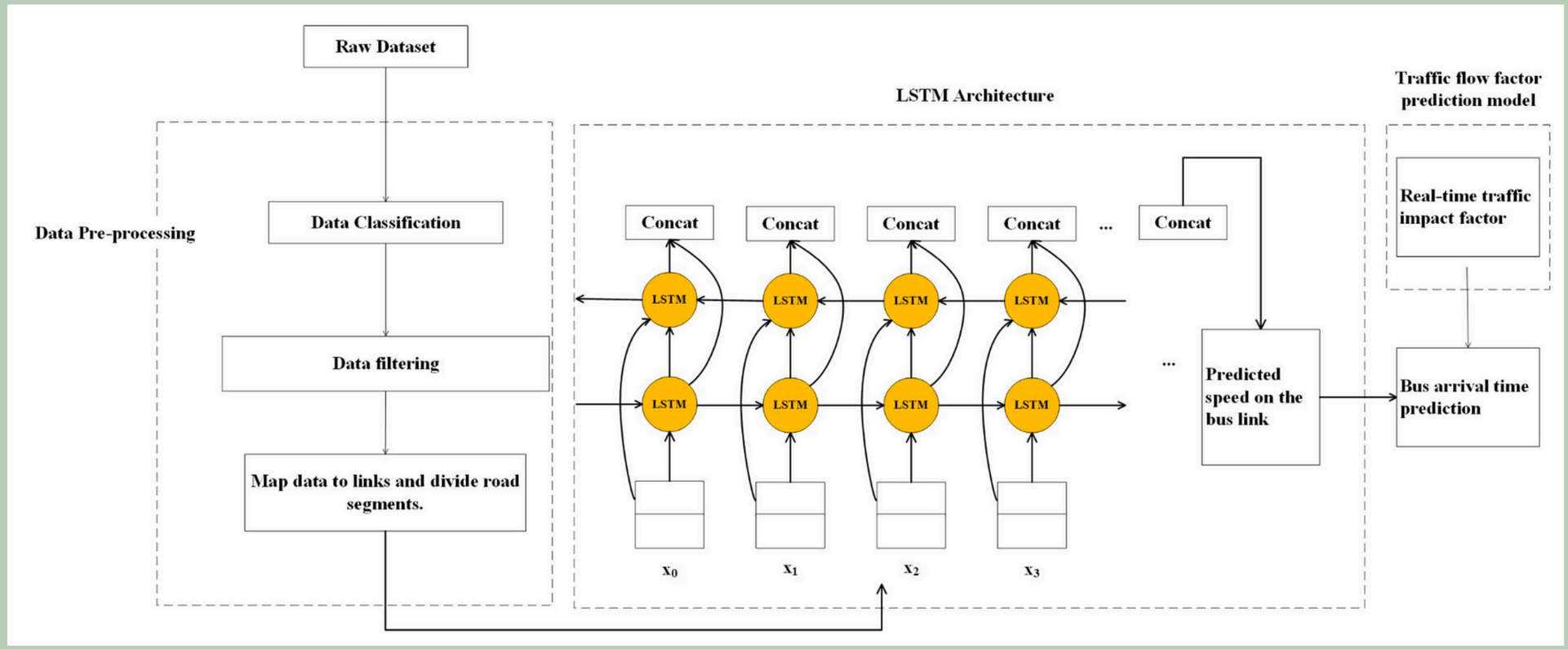


Fig 4:system architecture [6]

Used models

The following model were chosen to be applied on the data from Ingolstadt

Linear Regression

- Simplicity and Interpretability
- May not capture non-linear relationships, but helps in validating the features

Multi-Layer Perceptron (MLP)

- Handling Non-Linear Relationships
- commonly applied in practice to accomplish tasks of classification, regression, and forecasting

Long Short-Term Memory (LSTM)

- Capturing Temporal Dependencies
- Handling Non-Linear Relationships

Data we have

- FCD data as DataFrames

Column Name	Description
run	A unique identifier for each bus run.
utcTime	The timestamp of when the data was recorded in UTC.
offset	An offset value.
speed	The speed of the bus at the time of recording (in meters per second).
longitude	The longitude coordinate of the bus location.
latitude	The latitude coordinate of the bus location.
link	The identifier for the road segment or link the bus is currently on.
linkPos	The position of the bus on the current link, measured in meters from the start of the link.
route	The route number the bus is serving.
vehicle	A unique identifier for the bus vehicle.
trip	A unique identifier for the bus trip.
geometry	The geometric location of the bus represented as a POINT (in meters Easting and Northing).

Table 1:FCD data from bus

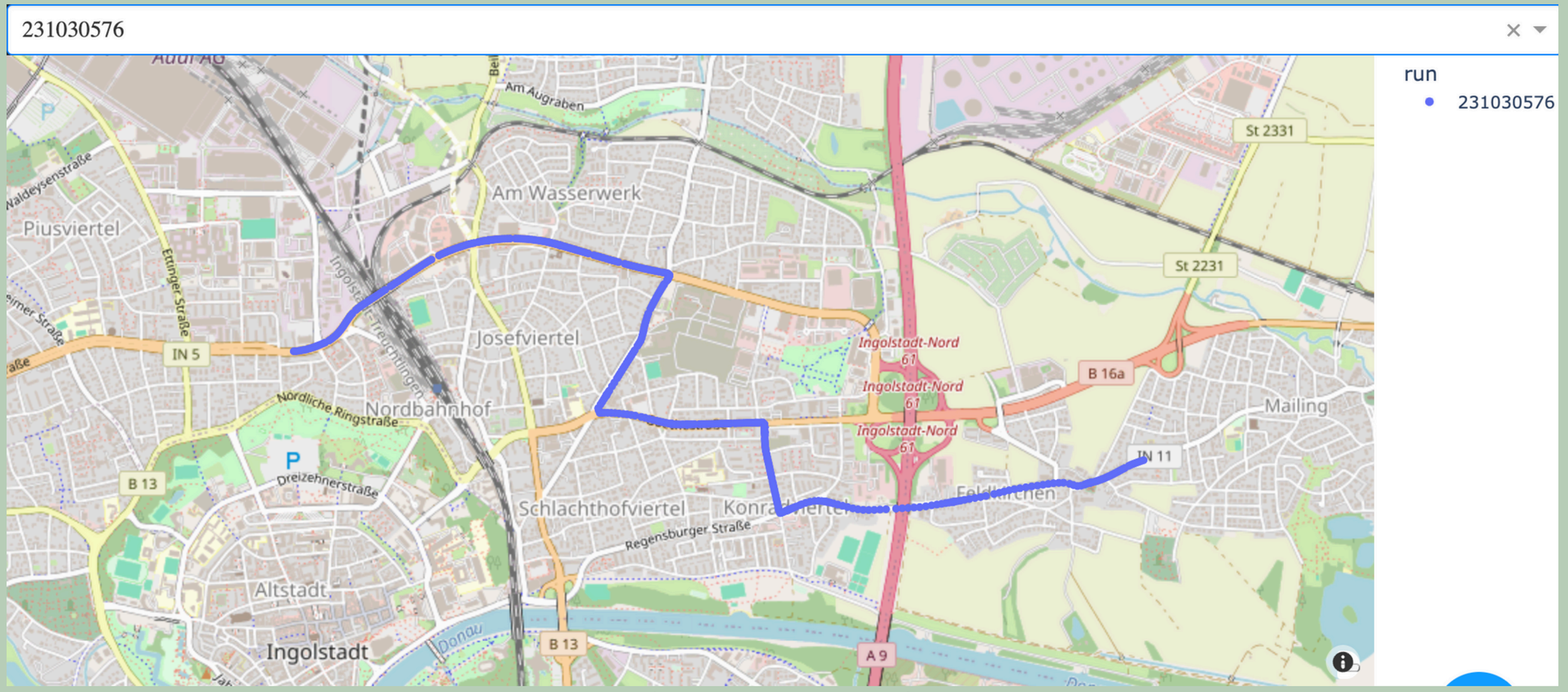


Fig 5:Run example

Data Preprocessing

Steps

Removing invalid runs and keeping runs in the considered direction

Considering congestion before traffic lights

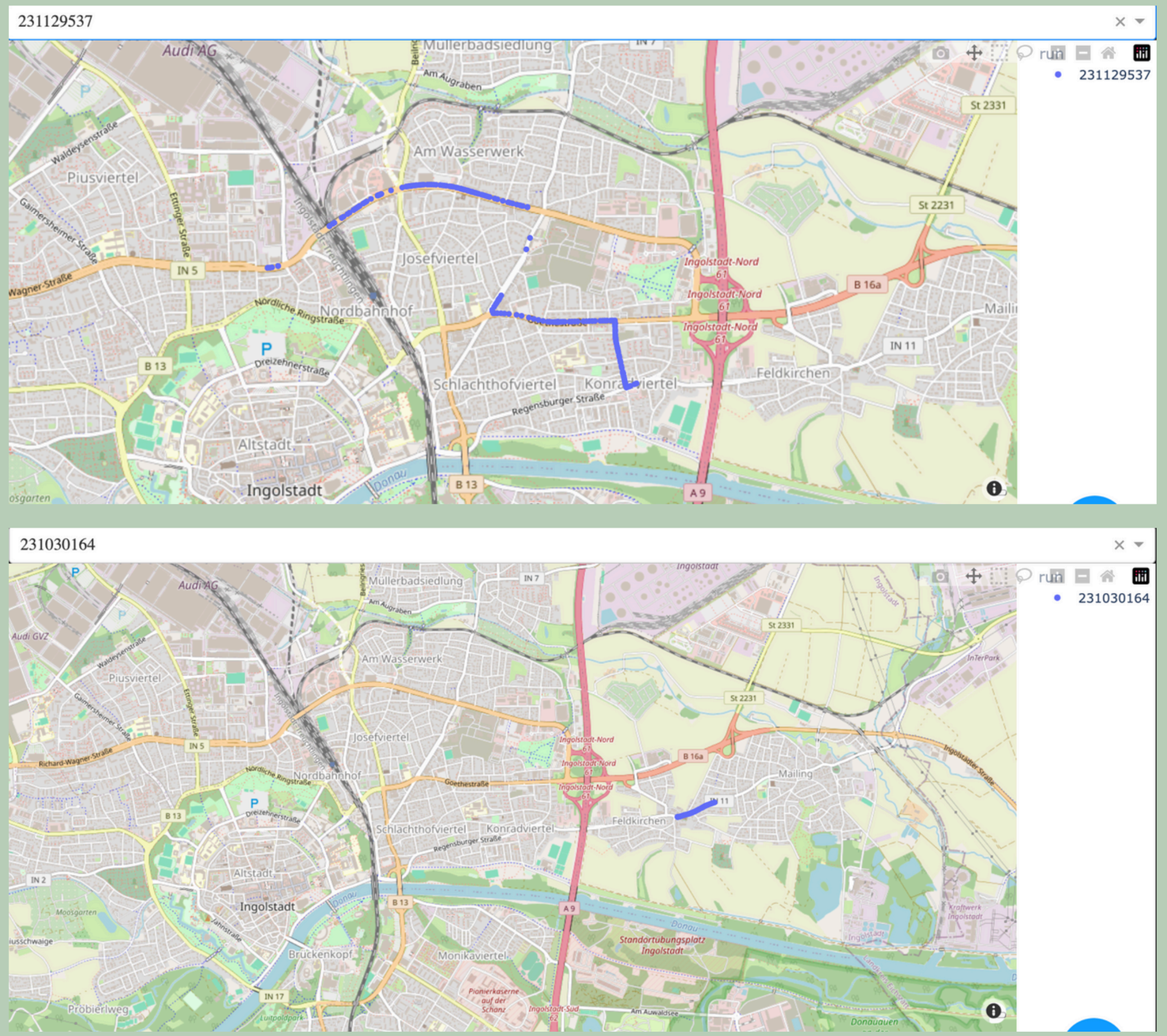
- 32s needed on average to cross 50m before stop line

Cut runs to specific distances

- 650m
- 350m
- 160m

Remove Bus stops waiting time for special use cases

Fig 6:Invalid run examples



Considering congestion before traffic lights

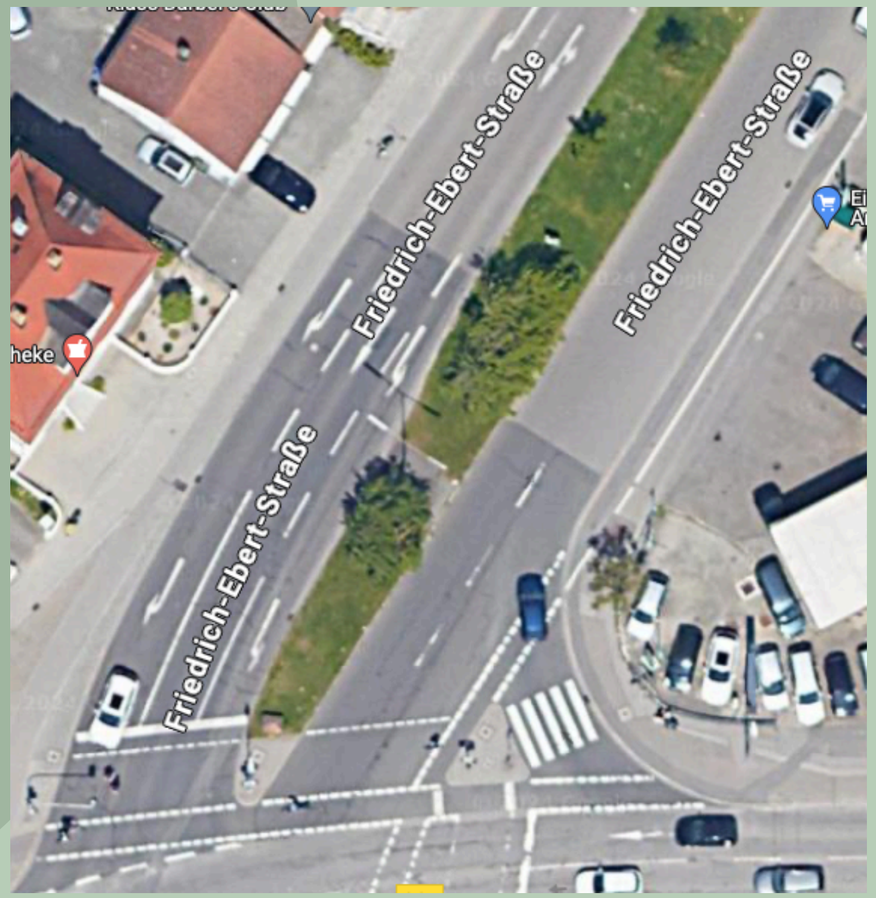


Fig 7:Traffic light

After cut

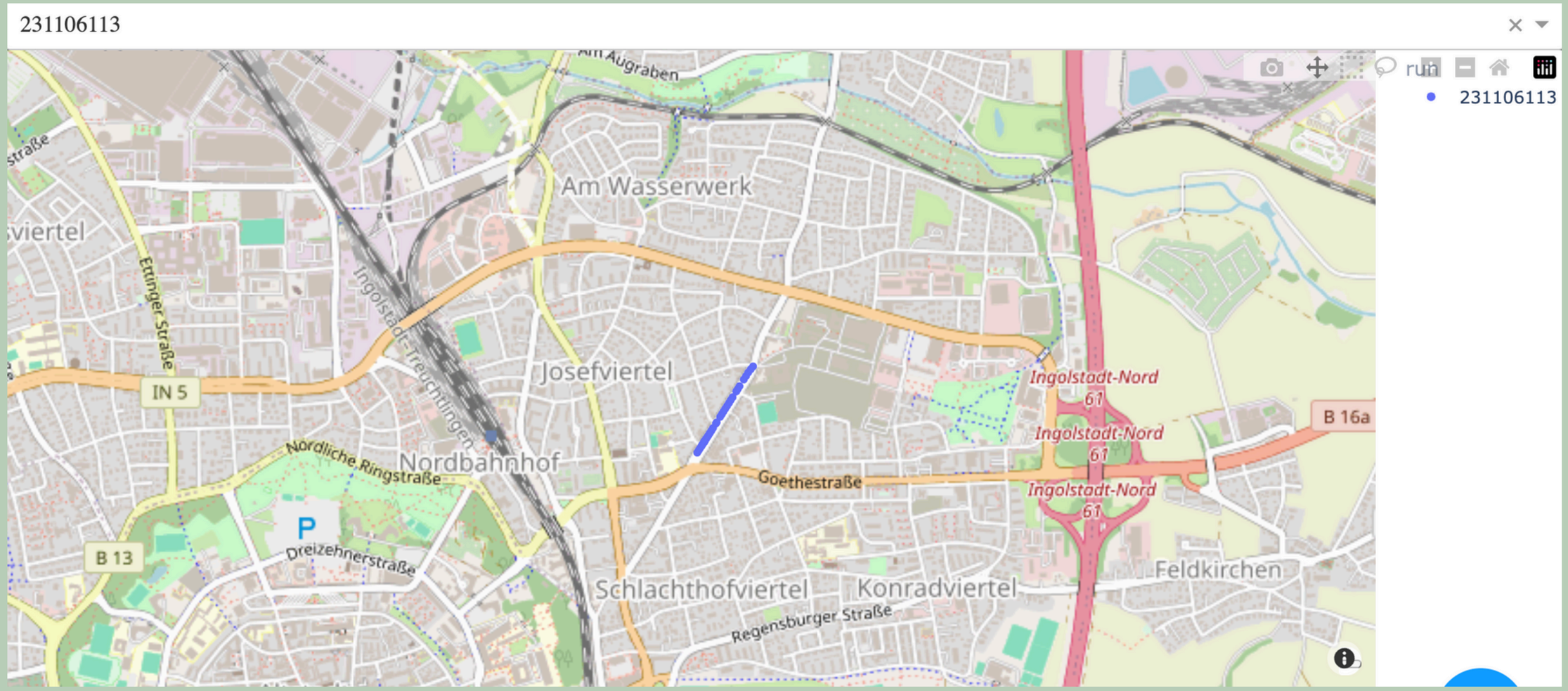


Fig 8:350m run

- Original number of runs: 24949
- After preprocessing: 11875

Exploratory Data Analysis

for 350 meters before the stop line

Points map with speed

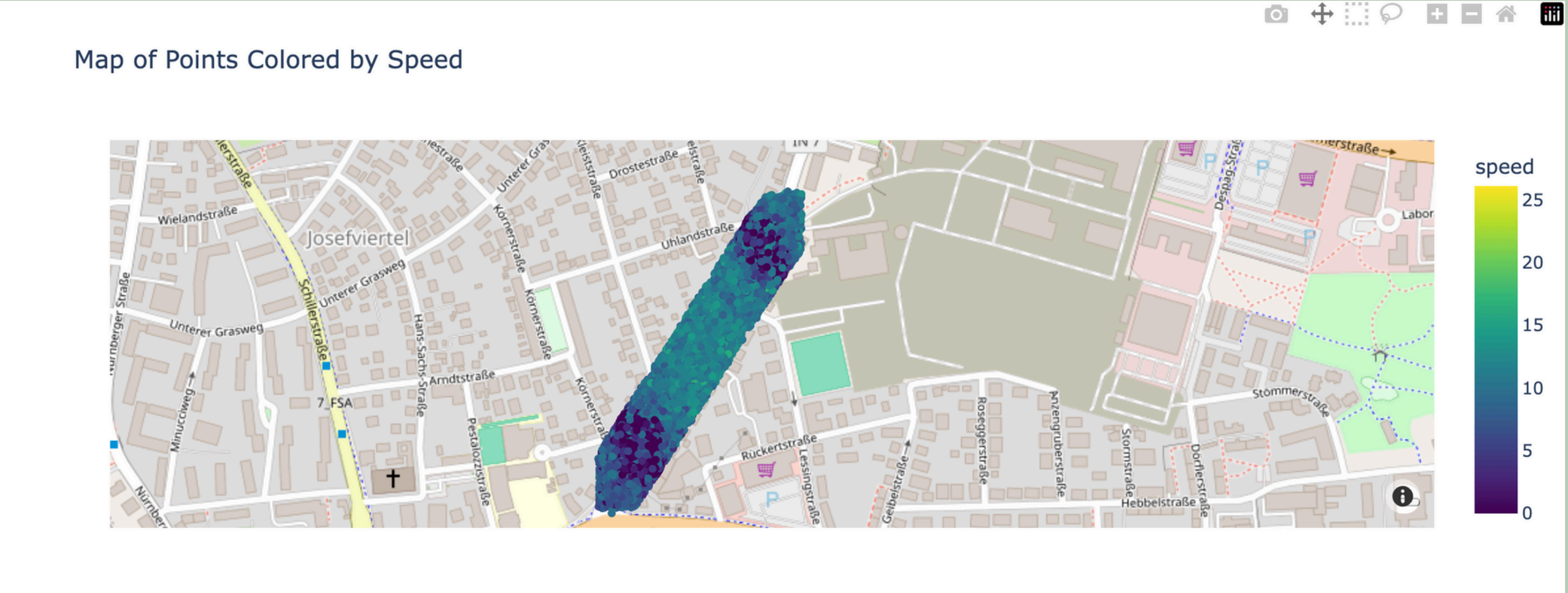


Fig 9: points map with speed indicator

- 41% of the buses stopped once and 35% stopped twice.
- Not all the runs stop

Run Duration Analysis

Distribution per Hour

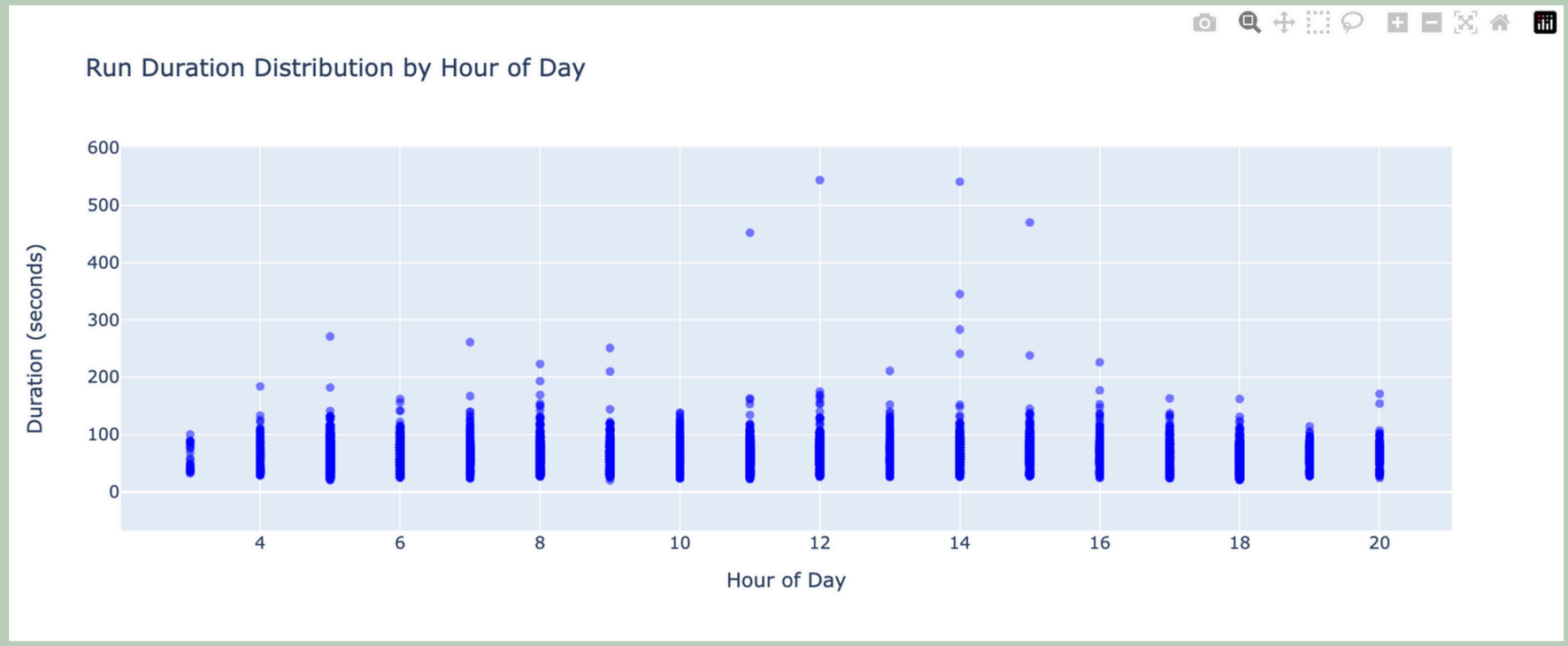


Fig 10:Distribution per Hour

- Consistency in Run Durations
- Presence of Outliers

Implication: Outliers should be removed

Average per Hour

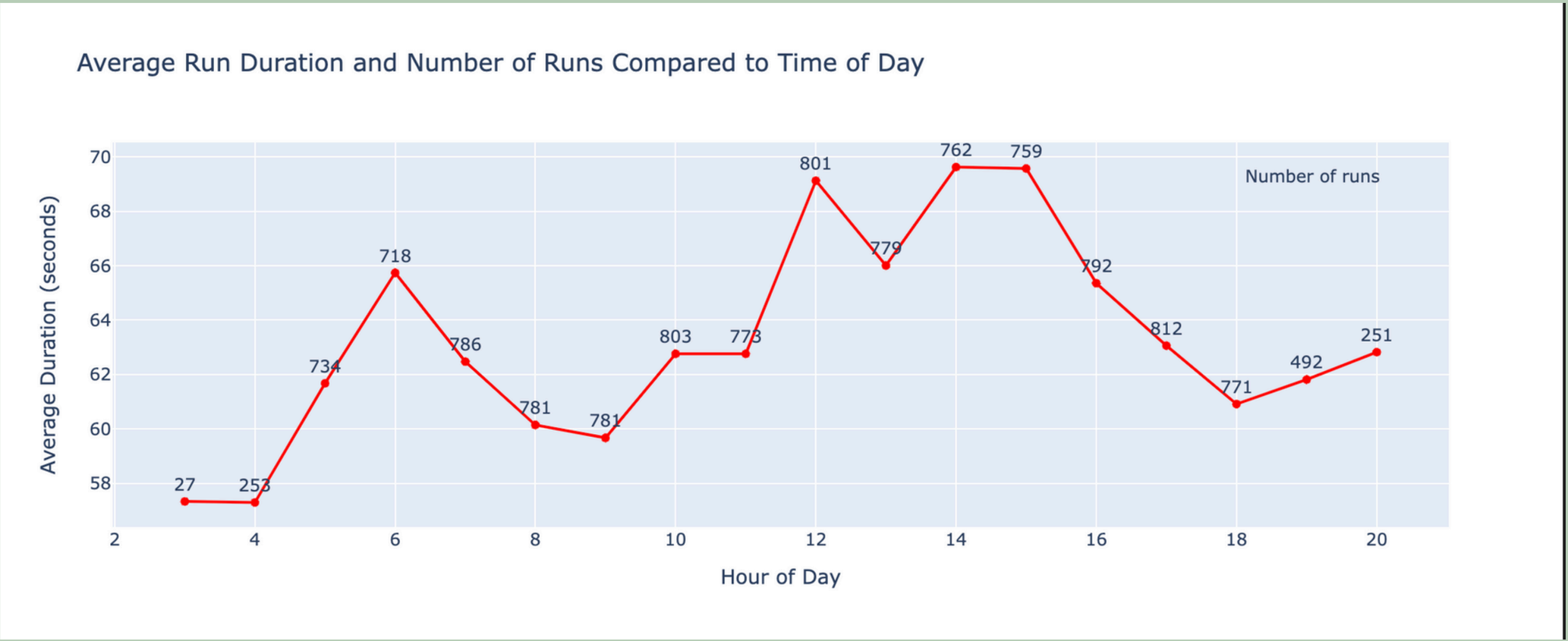


Fig 11:average per Hour

- Morning and Afternoon Peaks: noticed longer run times around 6 AM and 2 PM.
- Consistent Number of Runs: The most runs happen in the early morning and late afternoon.

Implication: The need to add the “time of day” in the model.

Distribution per Day

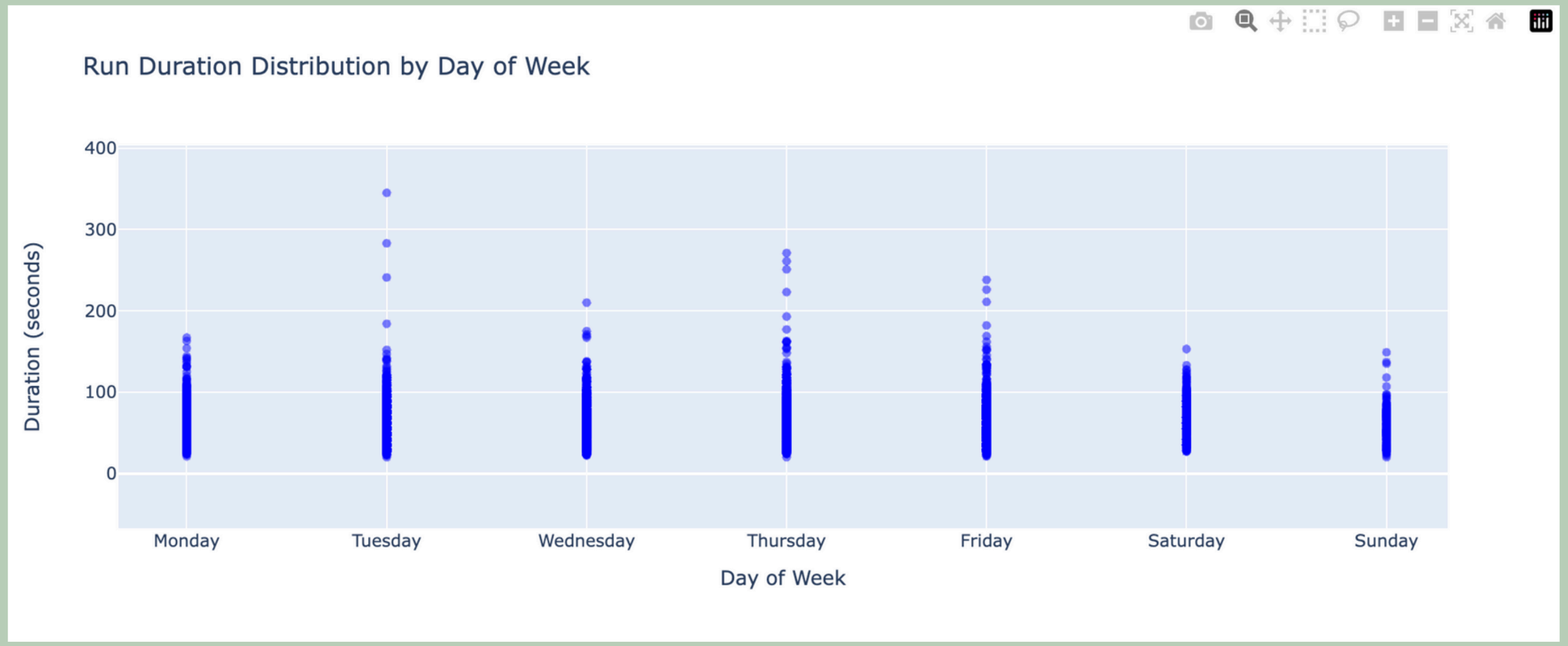


Fig 12:Distribution per Day

- Clear difference between weekdays and weekend
- Presence of Outliers

Implication: Outliers should be removed

Average per Day

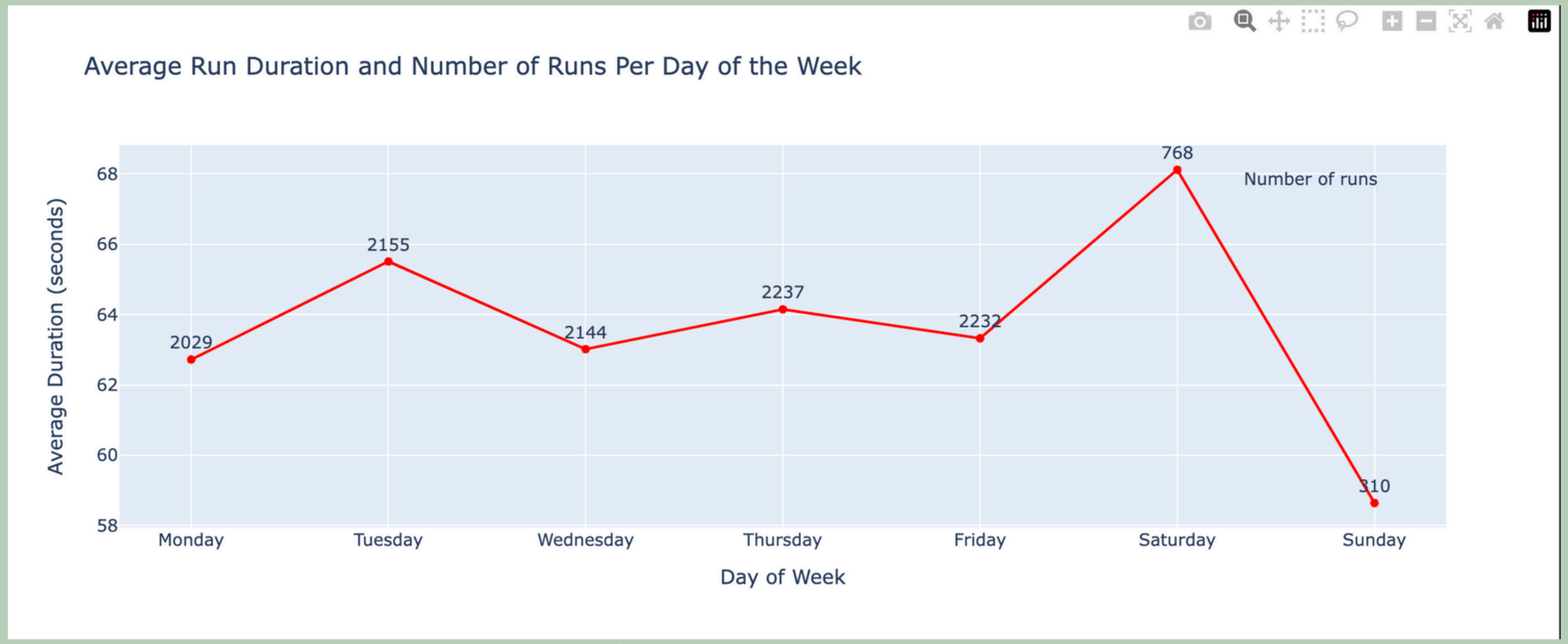


Fig 13:average per Day

- Almost consistent Average during the weekdays, the highest average for Saturday as everyone goes out, and the lowest average on Sunday as people prefer to stay home.
- Consistent Number of Runs on weekdays as bus comes every 10 mins, lower on Saturday as it comes every 30 mins, and the lowest on Sunday as it comes every one hour.

Implication: The need to add the “day of week” in the model.

Distribution per Week

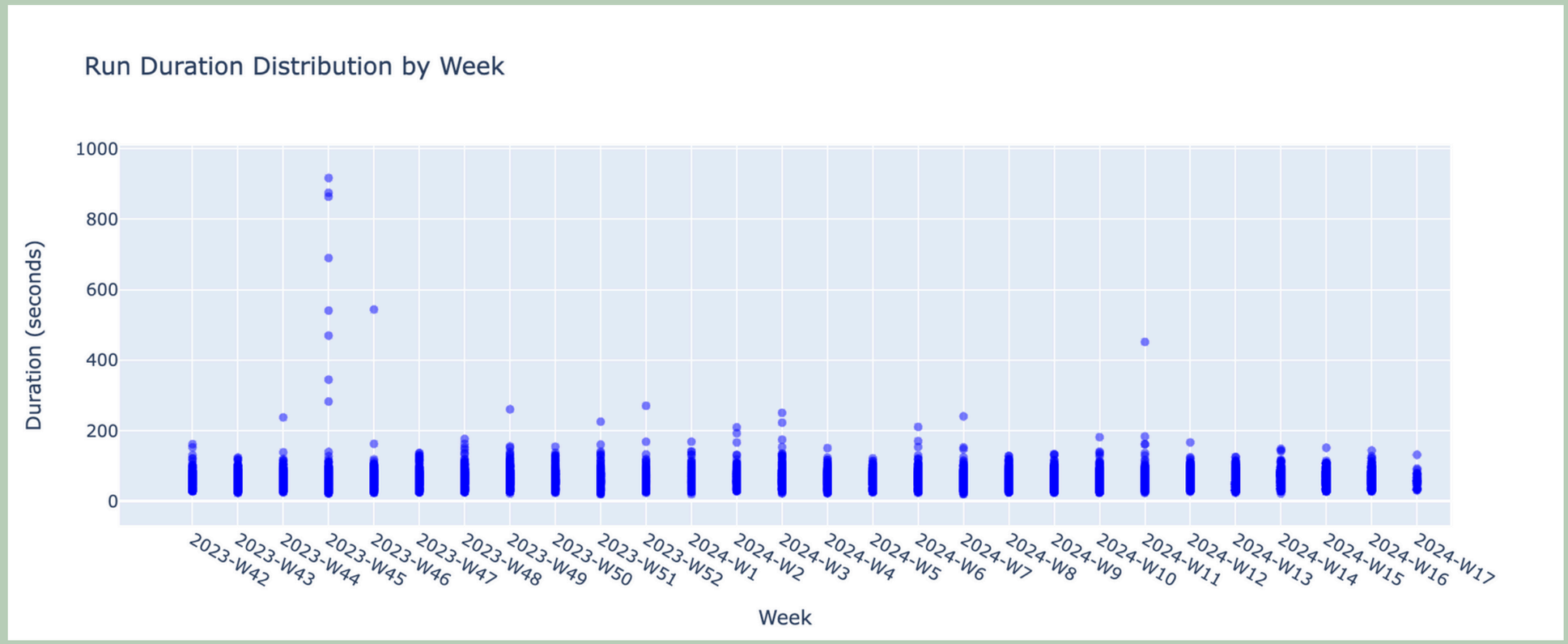


Fig 14:Distribution per week

- Consistency in Run Durations: Similar to the hourly distribution
- Presence of Outliers

Implication: Outliers should be removed

Average per Week

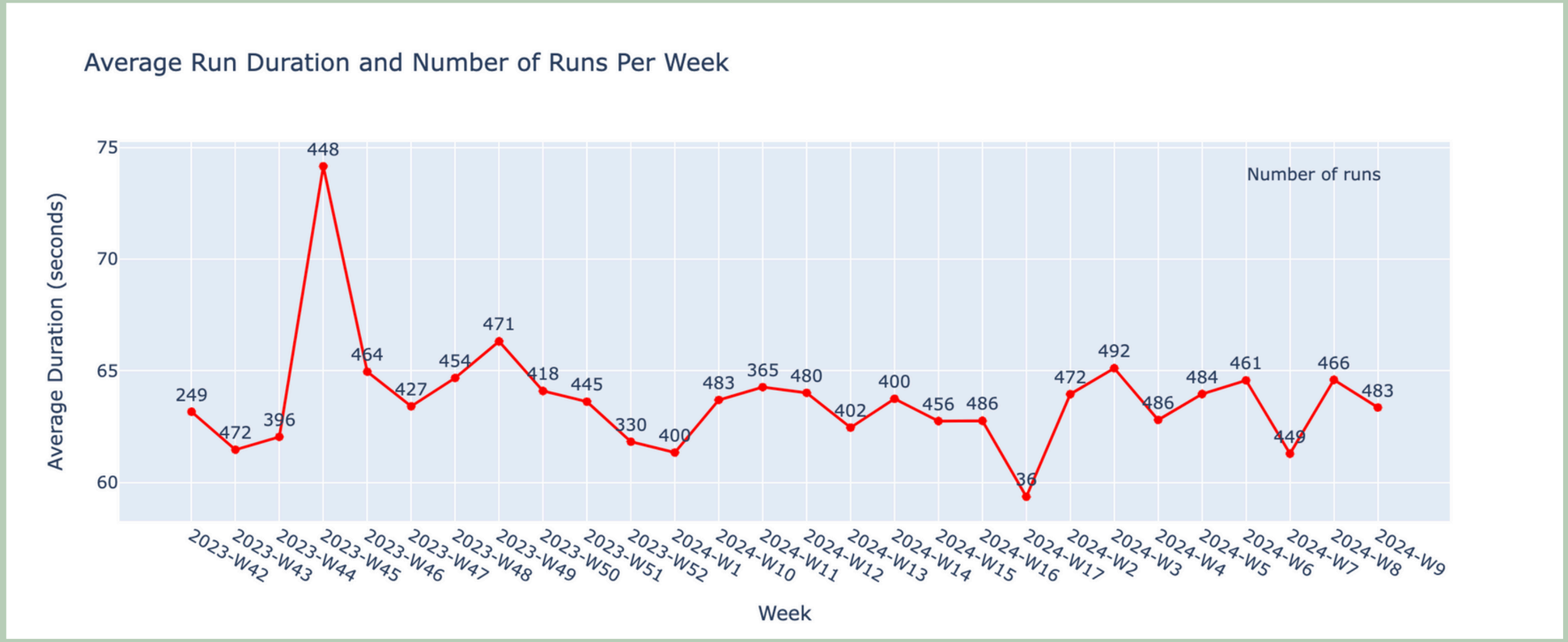


Fig 15:average per week

- Consistency with Fluctuations: There’s a peak in run times around week 45, but it smooths out after that.
- The number of runs stays pretty steady from week to week.

Implication: adding weekly trends to the model is needed.

Distribution per Month

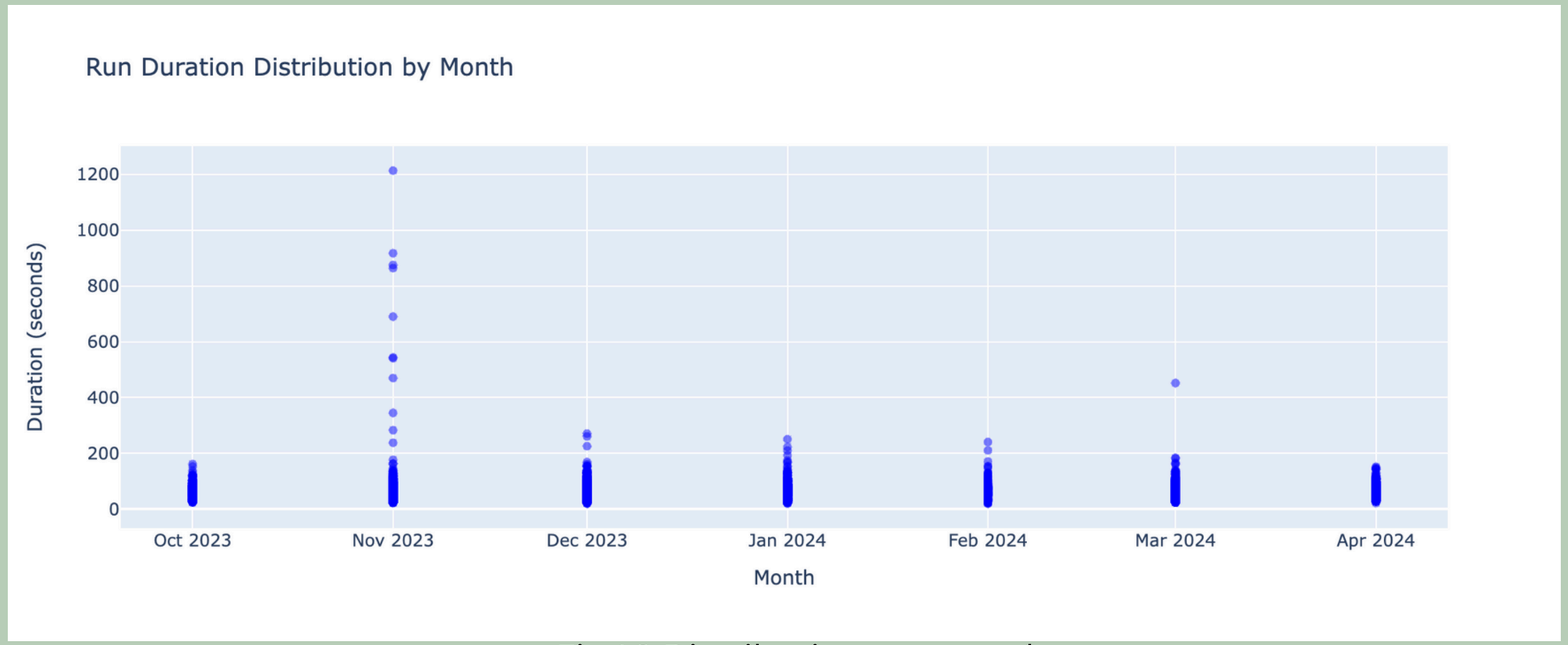


Fig 16:Distribution per month

- Consistency in Run Durations: The majority of the run durations are clustered between 50 and 100 seconds across most months.
- Presence of Outliers

Implication: Outliers should be removed

Average per Month

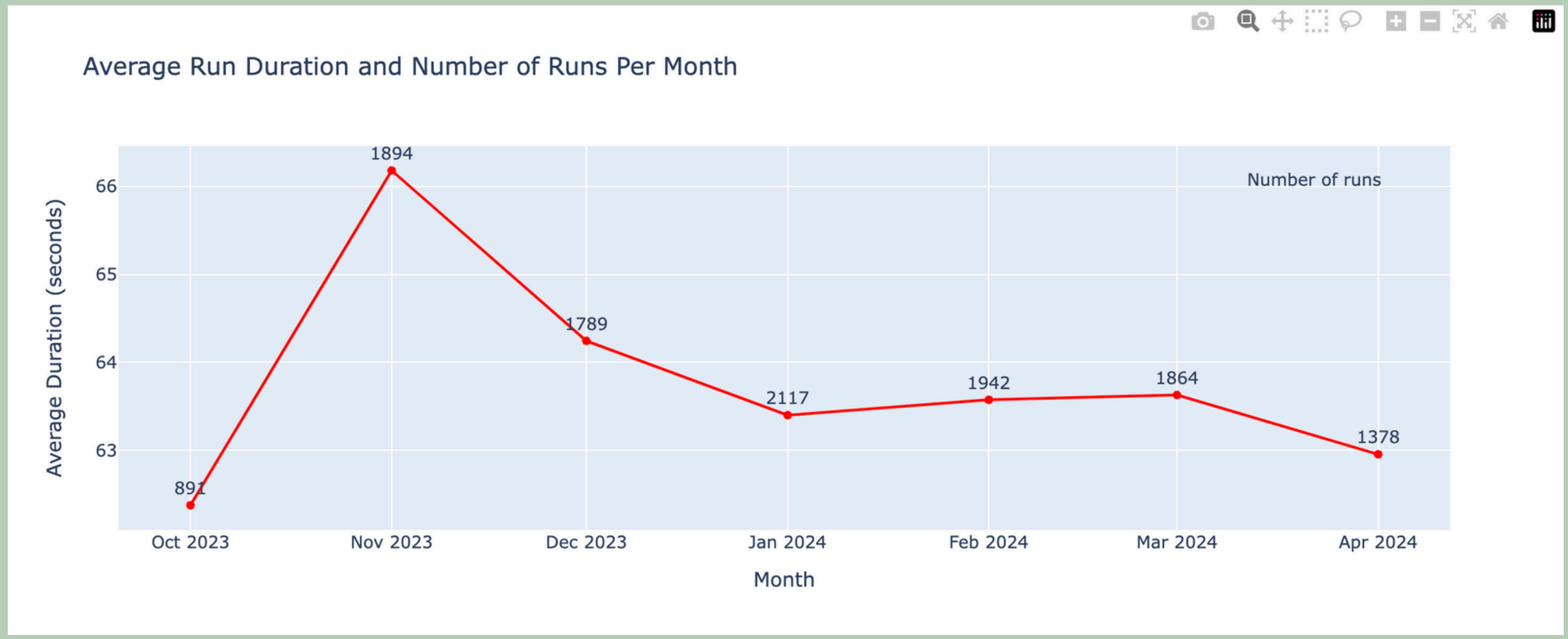


Fig 17:average per month

- November Peak: November has the longest run times, which then decrease in the following months.
- The number of runs is high in November but the highest in January.

Implication: I should add monthly trends to the model.

Descriptive Analysis

Statistic	Value
Count	11,875
Mean	63.91 seconds
Std Dev	28.09 seconds
Min	20.00 seconds
25% (Q1)	52.00 seconds
Median (Q2)	61.00 seconds
75% (Q3)	76.00 seconds
Max	1213.00 seconds
Variance	789.03
Range	1193.00 seconds

Table 2:Descriptive analysis

Defining the Outliers

- Using the Interquartile Range IQR

Given Values:

- Q1 (25th percentile): 52.00 seconds
- Q3 (75th percentile): 76.00 seconds

Calculate the Interquartile Range (IQR):

- IQR= Q3−Q1
- IQR= 76.00−52.00
- IQR= 24.00 seconds

Calculate the Lower Bound:

- Lower Bound=Q1−1.5×IQR
- Lower Bound=52.00−1.5×24.00
- Lower Bound= 16.00 seconds

Calculate the Upper Bound:

- Upper Bound=Q3+1.5×IQR
- Upper Bound=76.00+1.5×24.00
- Upper Bound= 112.00 seconds

Models & Results

Mean Model

- Used as a baseline
- In the most basic way

RMSE: 19.7 seconds

R-squared: 0.43

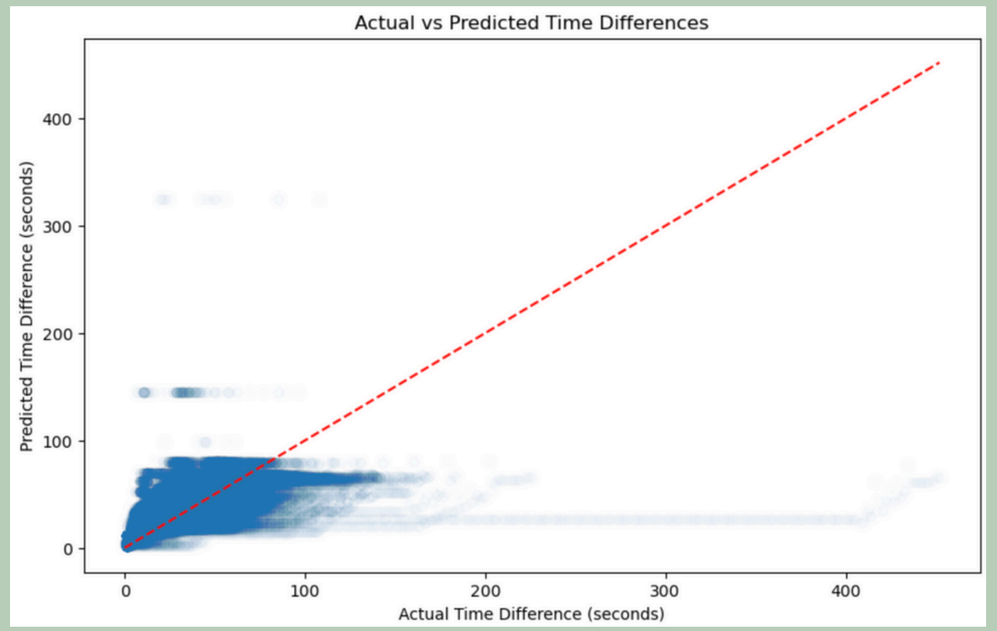


Fig 18:Actual vs Predicted

Without Dwell time

RMSE: 5.55 seconds

R-squared: 0.86

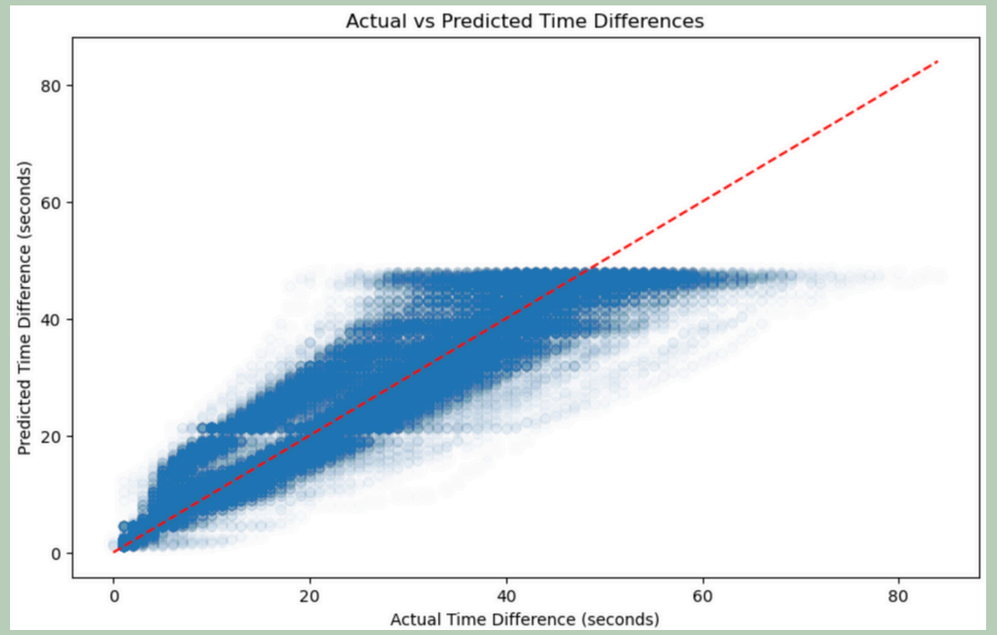


Fig 19:Actual vs Predicted

Linear Model

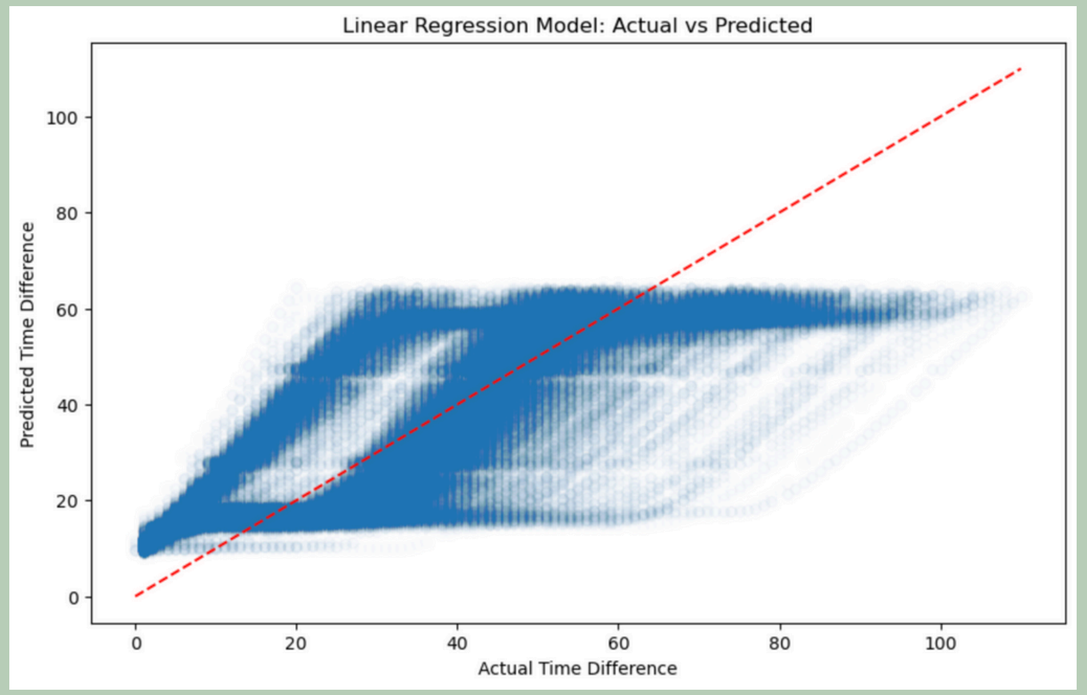


Fig 20:Actual vs Predicted

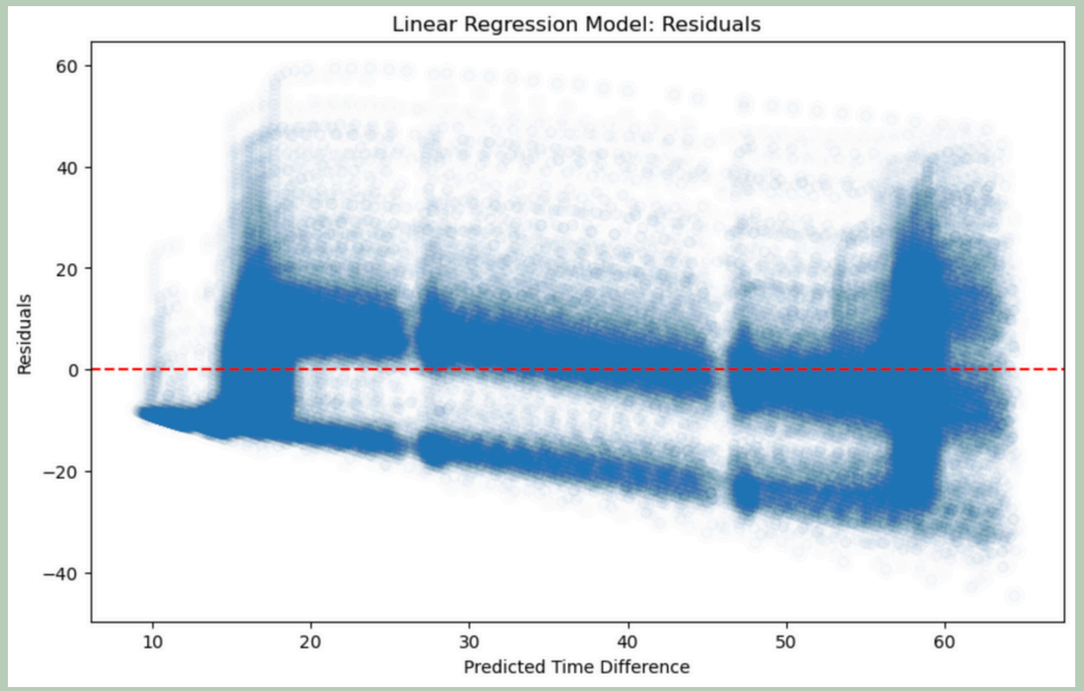


Fig 21:Residuals

RMSE: 12.4

R-squared: 0.67

Features evaluation

Using the coefficient value and the Variance Inflation Factor of the features

Features

- Distance
- Time of day
- Day of week
- Month of year

Feature	VIF	Coefficients
Distance	1.000085	17.99
day of week	1.008825	0.317
time of day	1.000230	0.319
month of year	1.008753	-0.229

Table 3:VIF and Coefficient value of Features

Without Dwell time

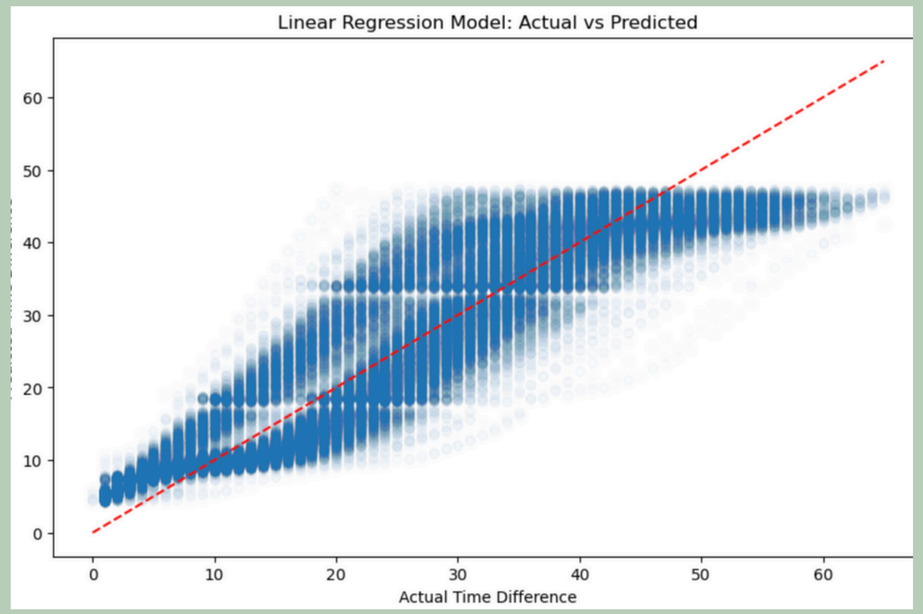


Fig 22:Actual vs Predicted

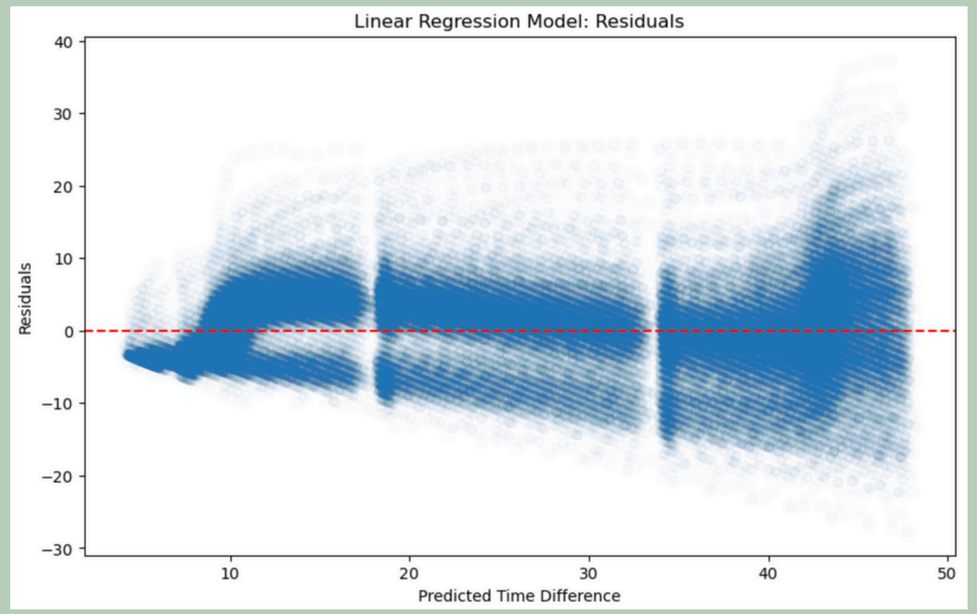


Fig 23:Residuals

RMSE: 5.5

R-squared: 0.86

MLP Model

- Input Layer: Corresponding to the number of features in the dataset.
- Hidden Layers: two layers with ReLU activation.
- Used layers: 16 neurons and 8 neurons.
- Output Layer: Single neuron for the regression output.
- Dropout Layer: to prevent overfitting
- Loss Function: Mean Squared Error (MSE)
- Adam optimizer

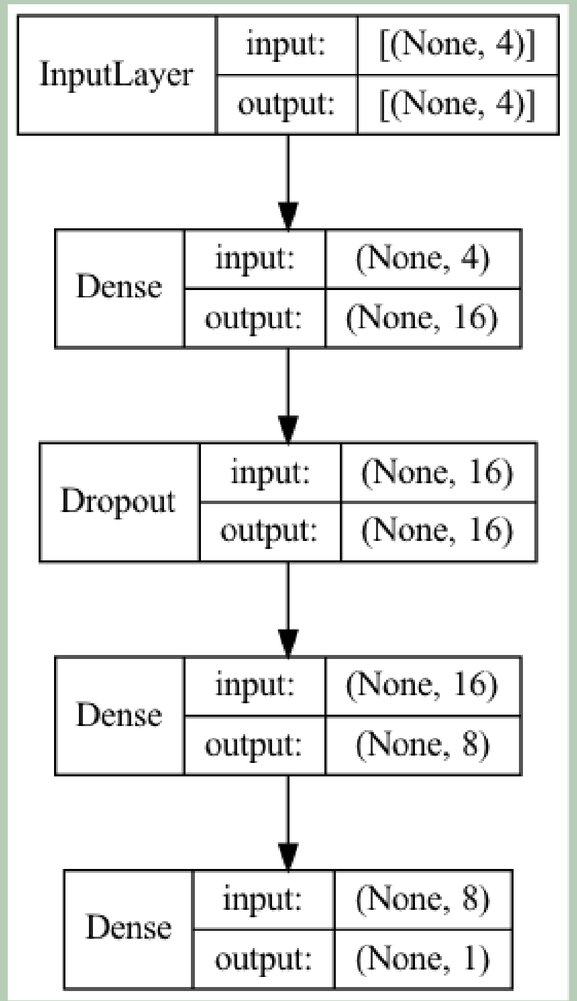


Fig 24:Model Structure

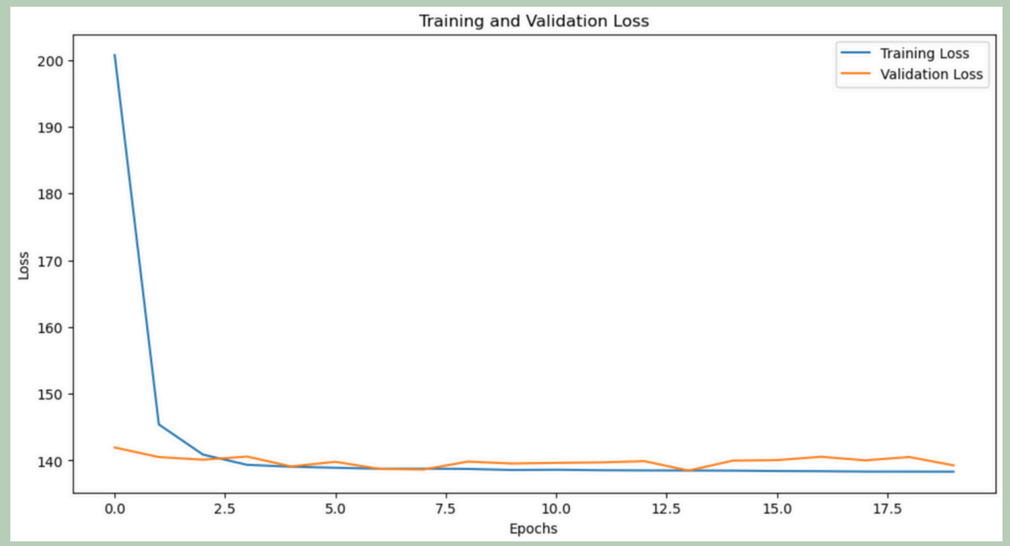


Fig 25:Training and Validation Loss

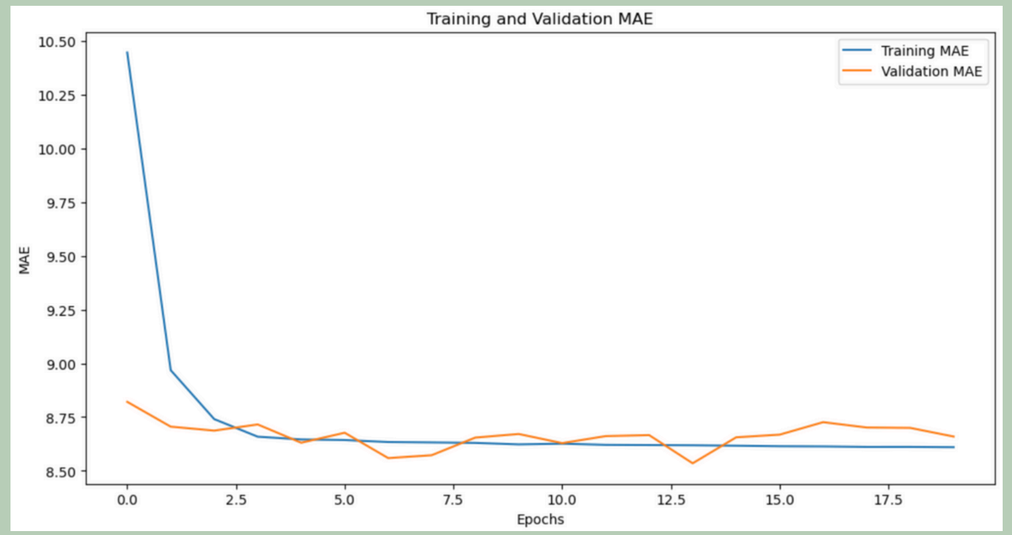


Fig 26:Training and Validation MAE

RMSE: 11.8

R-squared: 0.70

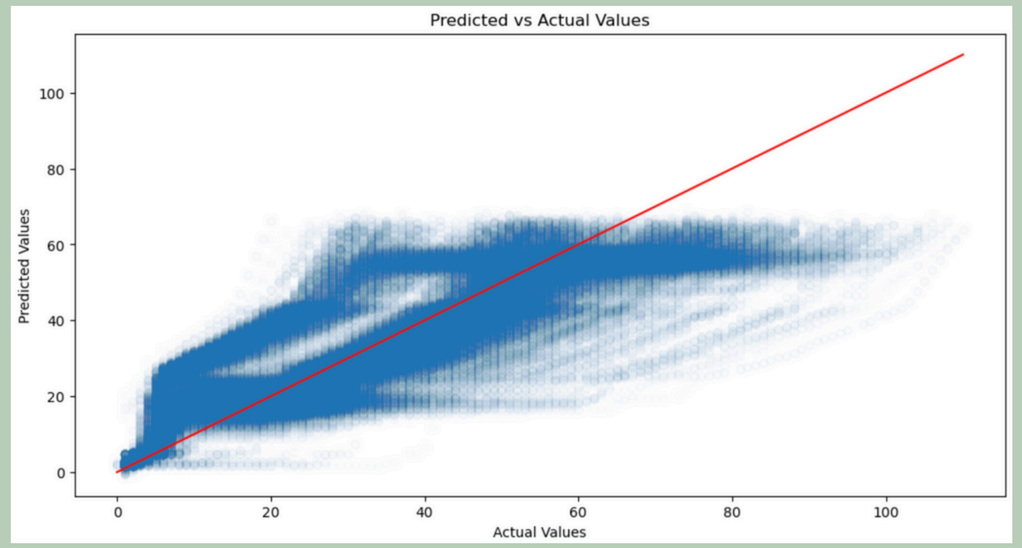


Fig 27:Predicted vs Actual Values

Without Dwell time

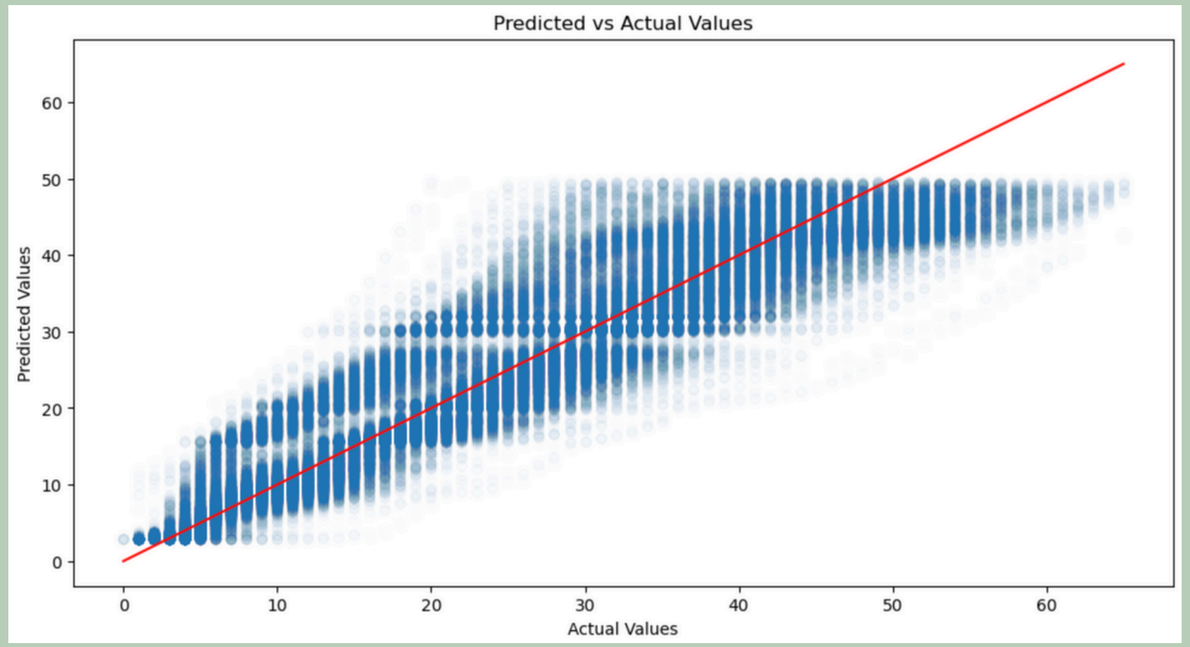


Fig 28:Predicted vs Actual Values

RMSE: 5.2

R-squared: 0.86

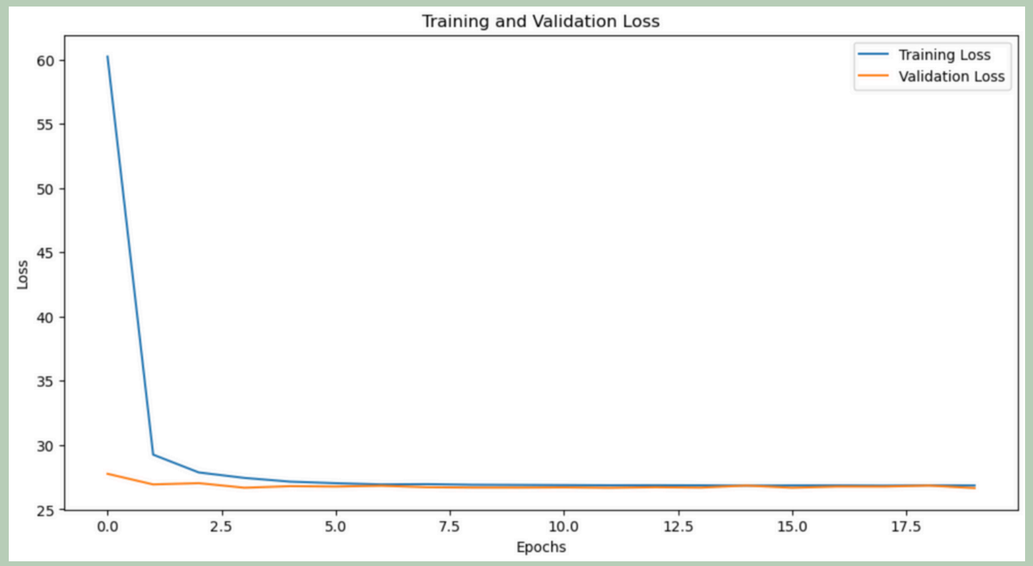


Fig 29:Training and Validation Loss

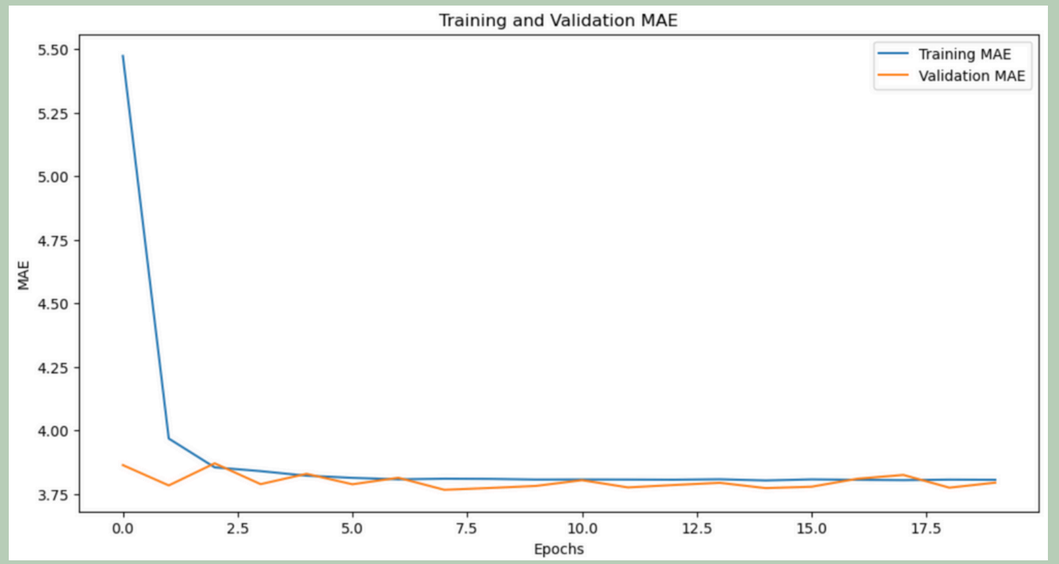


Fig 30:Training and Validation MAE

LSTM Model

- Input layer: 4 features
- LSTM layer: 16 cells with tanh
- Dropout: 16 cells
- Dense layer: 8 cells with ReLU
- Output layer: 1 cell
- Adam optimizer

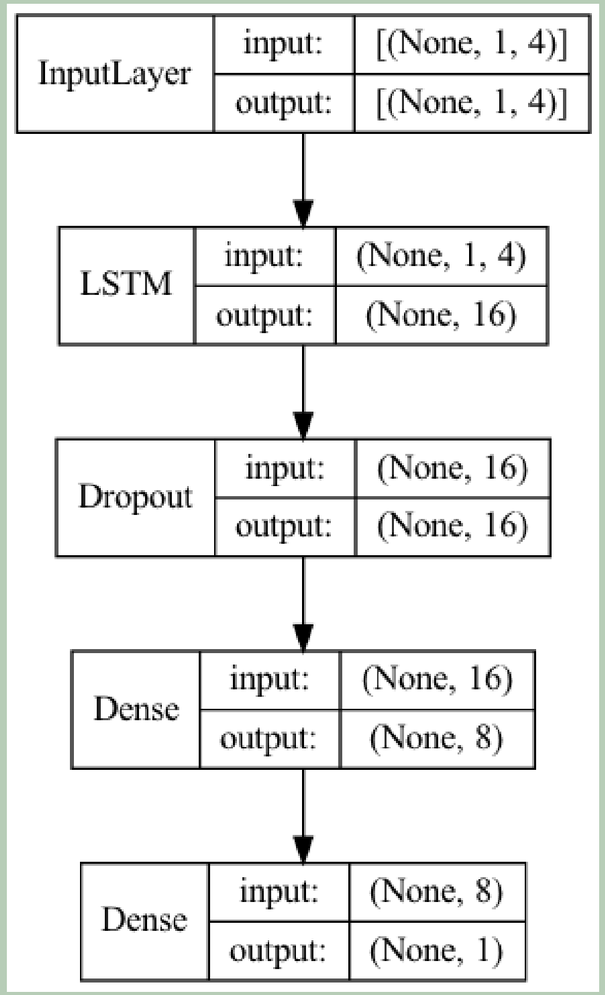


Fig 31:Model Structure

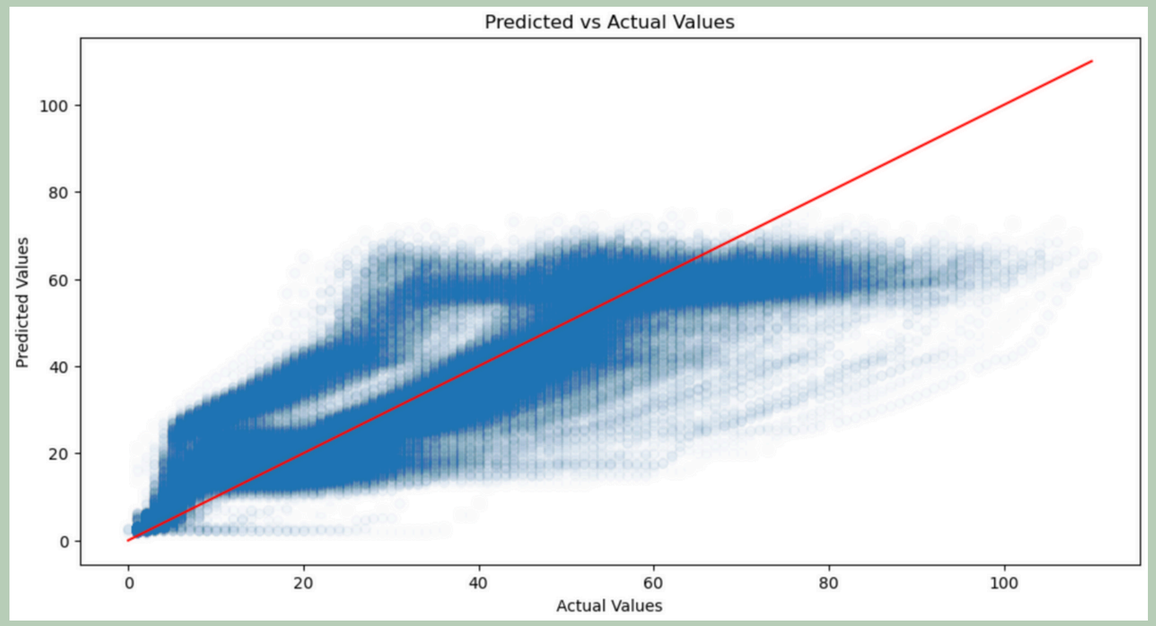


Fig 32:Predicted vs Actual Values

RMSE: 11.7

R-squared: 0.71

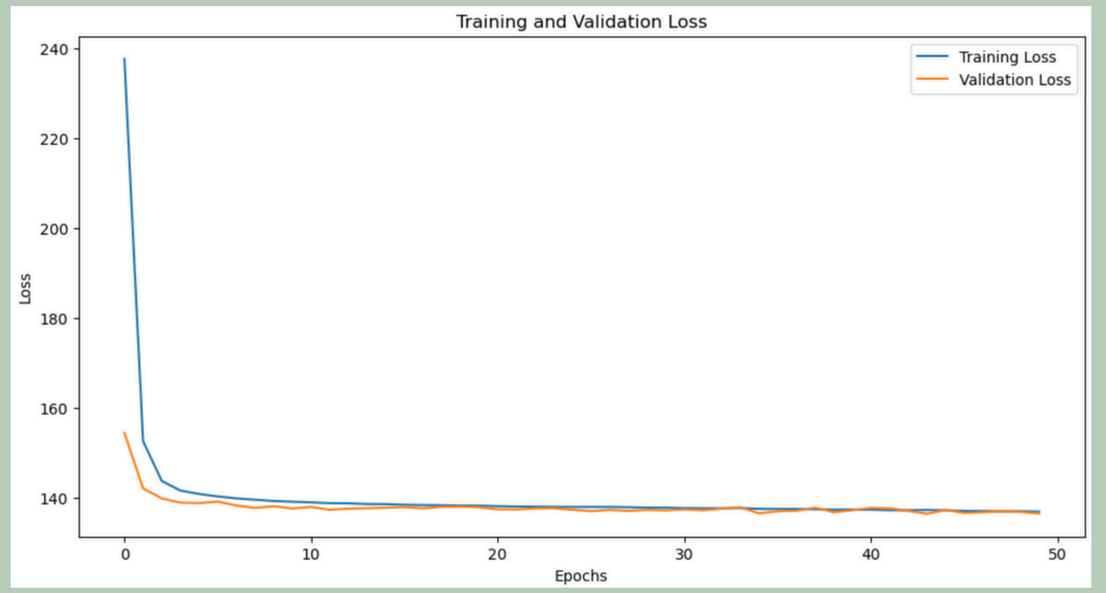


Fig 33:Training and Validation Loss

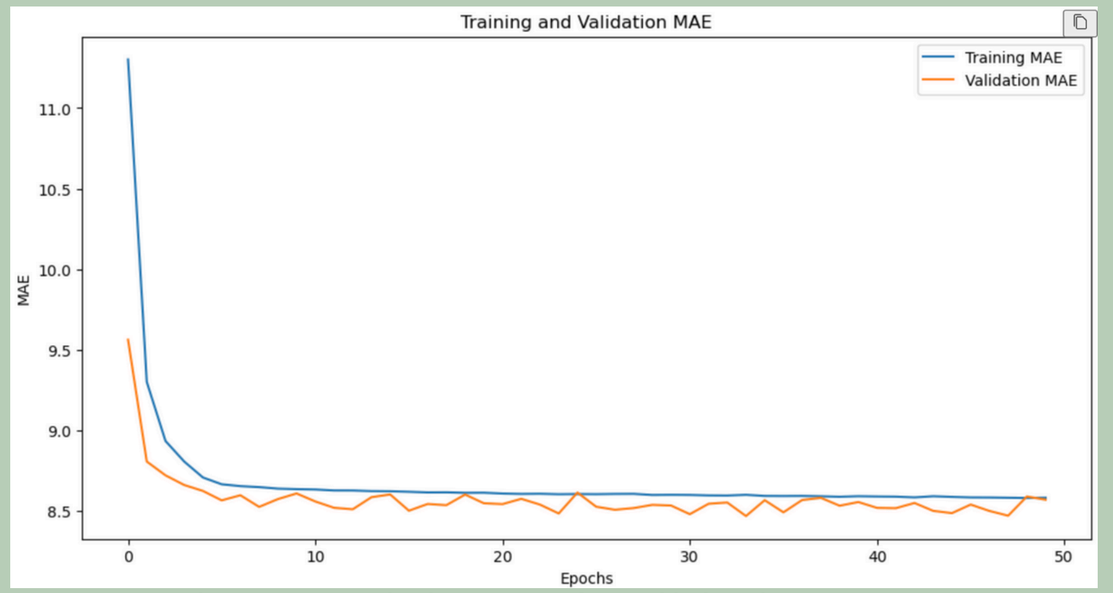


Fig 34:Training and Validation MAE

Without Dwell time

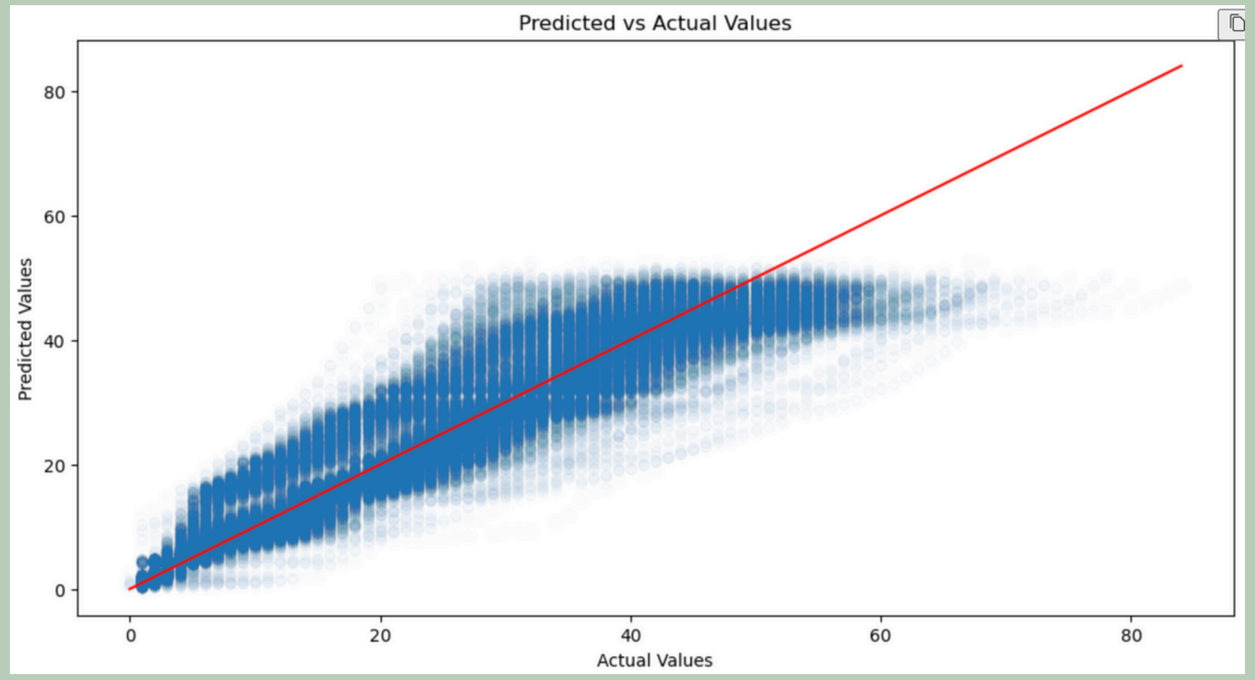


Fig 35:Predicted vs Actual Values

RMSE: 5.5

R-squared: 0.86

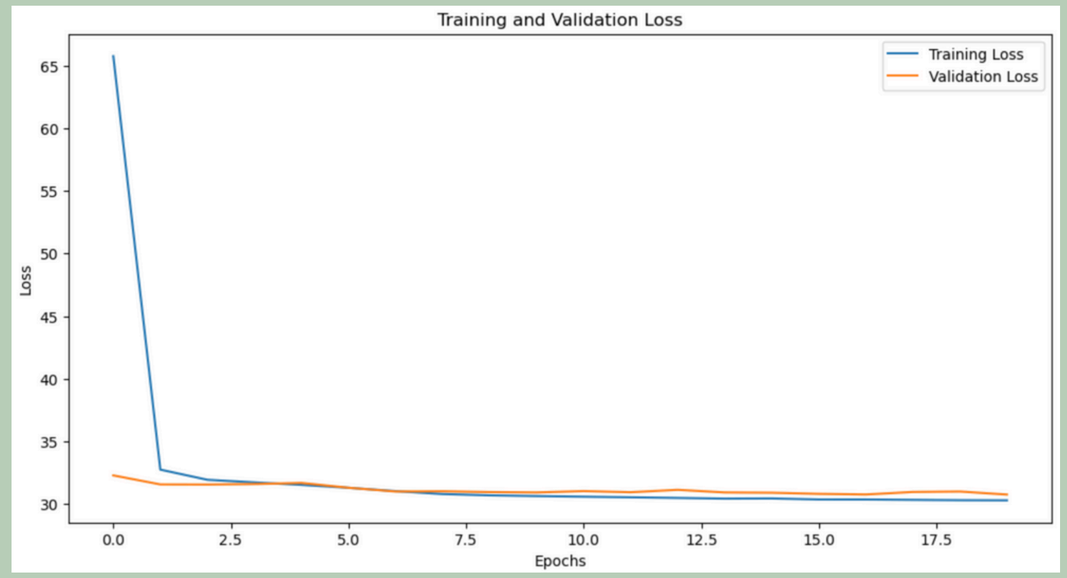


Fig 36:Training and Validation Loss

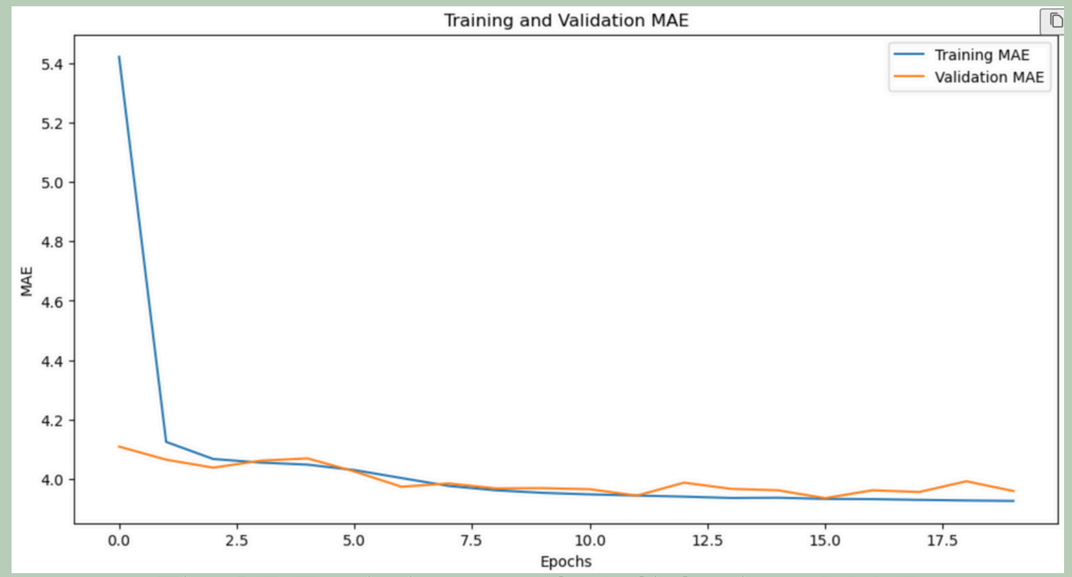


Fig 37:Training and Validation MAE

Models Comparison

Model	RMSE	R-squared
Mean	19.47	0.43
Linear Regression	12.4	0.67
MLP	11.8	0.70
LSTM	11.7	0.71

Table 4:Models with dwell time results

Without dwell time

Model	RMSE	R-squared
Mean	5.55	0.86
Linear Regression	5.5	0.86
MLP	5.2	0.86
LSTM	5.5	0.86

Table 5:Models without dwell time results

Conclusion

Summary of Findings

- LSTM slightly outperformed MLP in predictive accuracy.
- Excluding dwell times improved model predictions significantly.
- Distance to stop line was the most impactful feature.
- All models performed similarly without dwell times (RMSE ≈ 5.5s, R-squared = 0.86).
- Models need more information as they converge fast.

Summary of Contributions

- Significant improvement with Linear Regression, MLP, and LSTM over mean-based calculations.
- Data Preprocessing: Key techniques included:
  -Deleting invalid runs
  -Managing dwell times carefully
  -Focusing on data 50-400 meters from traffic lights
  -Accounting for congestion effects
- Feature Engineering: Enhanced model performance with features like distance, time of day, day of week, and month of year.
- Comparative Analysis: Evaluated strengths and weaknesses of models in various conditions, providing insights for future research and practical applications.

Limitations

- Predictions are made for only one stop line, limiting generalization.
- Data was collected over seven months.
- Models did not account for external influences like weather, passenger load, or road incidents, affecting prediction accuracy.
- Lack of Real-Time Updates: Models relied on historical data and lacked real-time adaptability, reducing effectiveness in dynamic traffic scenarios.

Future work

- Use one-year data and multiple traffic lights to improve model generalization and robustness.
- Include weather conditions, passenger loading, and road incidents to enhance prediction accuracy.
- Develop models that learn from streaming data for better adaptability to dynamic traffic conditions.
- Combine the strengths of different machine learning techniques to create more resilient and accurate predictive models.
- Establish evaluation metrics and validation protocols for consistent comparison and improvement across studies.

References

[1]https://mobility.mit.edu/publications/2016/osullivan-uncertainty-bus-arrival-time-predictions-treating-heteroscedasticity
[2]https://www.sciencedirect.com/science/article/abs/pii/S1568494621005846
[3]Harriet R. Smith. Transit signal priority (tsp). May 2005. URL https://web.archive.org/web/20060923120521/http://www.fta.dot.gov/documents/TSPHandbook10-20-05.pdf. Accessed: 10.07.2024.
[4]A. Taparia and M. Brady. Bus journey and arrival time prediction based on archived avl/gps data using machine learning. 2021. URL https://ieeexplore.ieee.org/document/9529328. Accessed: 11.07.2024.
[5]B. Ng C. T. Lam and S. H. Leong. Prediction of bus arrival time using real-time on-line bus loca- tions, 2019. URL https://ieeexplore.ieee.org/document/8947251. Accessed: 10.07.2024.
[6]H. Qingwen Y. Lei L. Fengxi Z. Lingqiu, H. Guangyan and C. Lidong. A lstm based bus arrival time prediction method. IEEE, January 2019. URL https://ieeexplore.ieee.org/ document/9060238. Accessed: 11.07.2024.

Thank you!