Machine Learning for traffic flow prediction at different junctions

A Project Report submitted by

Cyril John Arickathil - M22RM210

in partial fulfillment of the requirements for the award of the degree of

MTech in Data and Computational Science

Indian Institute of Technology Jodhpur Data and Computational Science

August, 2023 - July, 2024

Declaration

I hereby declare that the work presented in this Project Report titled **Machine Learning for traffic flow prediction at different junctions – M.Tech**. Submitted to the **Indian Institute of Technology Jodhpur** in partial fulfilment of the requirements for the award of the degree of M.Tech (Masters of Technology), is a Bonafide record of the research work carried out under the supervision of **Dr. Ranju Mohan**. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Cyril John Arickathil

M22RM210

Certificate

This is to certify that the Project Report titled **Machine Learning for traffic flow prediction at different junctions**, submitted by **Cyril John Arickathil (M22RM210)** to the **Indian Institute of Technology Jodhpur** only for the award of the degree of **M.Tech in Robotics and Mobility Systems (RMS)** is a Bonafide record of the work under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Ranju Mohan

Acknowledgment

I would like to extend my deepest gratitude to **Dr. Ranju Mohan** for her invaluable guidance and support as my project supervisor. Her insightful feedback, unwavering encouragement, and extensive knowledge have been instrumental in the successful completion of this project. Dr. Mohan's dedication to excellence and her willingness to share her expertise have greatly enhanced my learning experience. I am truly grateful for her mentorship and the opportunity to work under her supervision. Her insights and expertise were crucial to the successful completion of this work.

I am also immensely grateful to my examiners, **Dr. Bhupendra Singh** and Dr. Riby Abraham Boby, for their thorough evaluation, constructive feedback, and suggestions, which greatly contributed to the enhancement of this project.

My heartfelt thanks go to the program coordinators, **Dr. Gourav Bhatnagar** and Dr. Dilpreet Kaur, for their unwavering support, efficient coordination, and assistance in all administrative matters, making this academic journey a smooth and enriching experience. Finally, I would like to acknowledge the support and encouragement from my family,my elder brother, friends, and colleagues, without whom this project would not have been possible.

Thank you,

Cyril John Arickathil

Abstract

This research focuses on the application of machine learning in optimizing public transport planning by developing models to predict traffic flow at junction levels in various conditions, particularly on freeways or highways.

The escalating challenge of urban traffic congestion underscores the critical need for a robust traffic monitoring and forecasting system, with accurate prediction of traffic flow being a core component. This study emphasizes the effectiveness of a proposed framework in understanding traffic flow waves and congestion at the junction level.

The findings highlight the necessity of identifying the most effective solution among Recurrent Neural Network (RNN) variants, including Long Short-Term Memory (LSTM) models, neural networks. These models exhibit superior performance in link flow predictions compared to other machine learning methods. The choice between RNN, or LSTM models is crucial for enhancing the accuracy and efficiency of traffic flow predictions.

Practical insights underscore the importance of the deep learning model structure, data preprocessing, and error matrices for achieving precise traffic flow predictions. Implementing the suggested approaches contributes to the optimization of traffic flow, thereby enhancing overall urban transportation efficiency and fostering a more sustainable and seamless urban mobility experience.

Contents

Chapter 1

Introduction

1.1 Background and context

In cities, traffic jams at stoplight intersections are a big problem that makes transportation systems work less well. More and more people moving into cities and more cars on the road make it even trickier to find ways to deal with this issue. We really need new and smart ideas to handle and improve the flow of traffic. Machine Learning (ML), a kind of technology that can predict things, makes up for a good solution. It can help make traffic management at stoplight intersections better by telling us what might happen ahead of time.

Lots of people living in cities and more cars on the roads mean more traffic jams at stoplights (traffic signals). The usual systems we use to control traffic struggle to adapt to these changes and don't always work well. That's why we need a different approach. Machine learning, which can learn from past data, seems like a good fit. By looking at how traffic behaved before, it can predict how it might behave in the future at stoplight intersections. Making decisions about traffic in real-time is super important, and machine learning helps us do it fast and smart. So, looking into how machine learning can predict traffic at stoplight intersections is important to create smarter and better ways to handle traffic in our cities. Various methodologies of machine learning and deep learning are already in use from which we can be able to draw some valuable insights on what can be the best option for us to implement for our problem statement.

Technology used for this project: predominantly for this project python with its libraries has been used for understanding the neural networks and modelling. PyTorch functions and its application will be discussed followed by the mathematical interpretation behind the same.

1.2 Research questions

RQ1: What are the use cases in which machine learning model can be utilized in traffic prediction for context where some level of history of congestion at a point in time is known?

RQ2: What is the algorithm to train the machine learning model to predict the outcome for a time-series data way ahead in time?

Chapter 2

Literature survey

2.1 Supervised Learning

The machine learning model built using existing dataset is called supervised learning. ´Figure1 illustrates the supervised learning workflow. In this paper, author builds a supervised learning model to predict the product taxonomies. The work flow starts with data collection, this step involves fetching the corpus of raw data. This raw data processed before creating a training data set out of it. The data preprocessing steps involves feature selection, text normalization, data imputation and feature extraction. The training data is passed an input to the model which compares its results with the actual value to self- evaluate and learn. If any hyperparameter tuning for the machine learning model is required then it is trained again with new parameters. The model is then evaluated and then deployed for production use.

2.2 Machine Learning Techniques in Traffic Prediction

Machine learning techniques have shown great promise in traffic flow prediction due to their ability to learn from data and capture complex patterns. Key machine learning techniques explored in the literature include:

- Linear Regression
- Decision Trees and Random Forest
- Support Vector Machines (SVM)
- Neural Networks
- Deep Learning Models (Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM))

2.3 Comparative Studies

Several comparative studies have evaluated the performance of different machine learning models in traffic flow prediction. These studies provide insights into the strengths and weaknesses of various models, guiding the selection of appropriate techniques for specific scenarios.

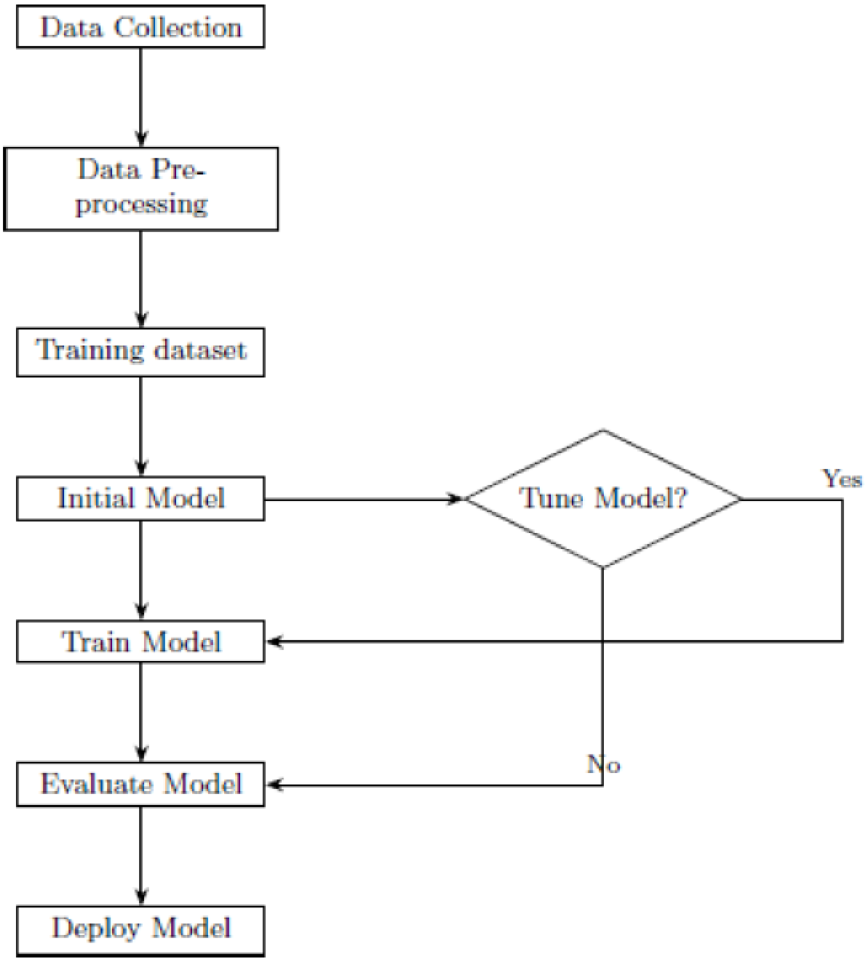


Figure 2.1: Supervised Learning

Chapter 3

Methodology

3.1 Start with small prototype

Predicting the number of vehicles on a particular junction involves processing timeseries data. Initially the author processes the data only forone of the junction. The raw data has been used from Kaggle.com provided by (fedesoriano, 2021).

About the dataset :
The dataset contains data from year 2015-2017 along with junctions and number of vehicles in hourly intervals. The data manipulation is performed using Python library-Pandas.

```
    df = pd.read_csv(’Data/traffic.csv’)
    df = df.drop(’ID’, axis=1) df.head()
```

Listing 3.1: Data set fetching using Pandas

```
    index,DateTime,Junction,Vehicles,ID
    0,2015-11-01 00:00:00,1,15,20151101001
    1,2015-11-01 01:00:00,1,13,20151101011
    2,2015-11-01 02:00:00,1,10,20151101021
    3,2015-11-01 03:00:00,1,7,20151101031
    4,2015-11-01 04:00:00,1,9,20151101041
```

Listing 3.2: Sample Result

At 4 junctions vehicles passing in every 1 hour recorded. Entire test dataset available in “traffic.csv” file

3.2 Time series extraction

Developing a machine learning model to predict the flow of traffic considering multiple aspects such as exact point of year(month-wise), also on the daily basis considering progression hour-wise is a complicated process.

We need to split the data such that `DateTime` column is split into ‘months’, ‘hours’, ‘day’. This is important as various aspects such as ‘day of the week’, ‘month in a year’, ‘hours in a day’ need to be checked against for validating the trend in traffic.

```
    df[’DateTime’] = pd.to_datetime(df[’DateTime’], format=’mixed’)
    # Exploring more features
    df_Junction1["Year"]= df_Junction1[’DateTime’].dt.year
    df_Junction1["Month"]= df_Junction1[’DateTime’].dt.month
     df_Junction1["Date_no"]= df_Junction1[’DateTime’].dt.day
     df_Junction1["Hour"]= df_Junction1[’DateTime’].dt.hour
    df_Junction1["Day"]= df_Junction1.DateTime.dt.strftime
    ("%A") df_Junction1.head()
```

Listing 3.3: Data set fetching using Pandas

3.3 Ideate: Data Exploration

Initial approach of finding a solution to predict time-series event is to analyze already existing solution to a different type of prediction problem.

For example, (**Goodfellow-et-al-2016**) showcases the implementation of RNN (Recurrent Neural Network) techniques in handling time-series events. In this context, the author leverages historical data patterns, to forecast the progression of traffic over time. The primary goal is to predict traffic conditions based on learned historical patterns.

In tailoring this methodology to the intricacies of traffic flow, the author recognizes the complexity of the task and opts to utilize past data patterns for making predictions.

To assess the effectiveness of the model, the author employs a visual tool, such as plotting graphs based on various traffic events. This visual representation offers a clear evaluation of predicted versus actual traffic conditions. Additionally, this approach allows for manual verification and scrutiny of instances where the model may have inaccurately predicted traffic conditions, contributing to a more comprehensive understanding of the model's performance.

3.4 Understanding Mathematics of Neural Networks

In this paper, author researches on the relevance of the mathematical concept with respect to the machine learning process. Understanding the math behind reducing the loss for prediction enables to completely understand the algorithms. Especially the training algorithm applies the activation functions. Author describes the theoretical knowledge with reasoning to apply these activation functions.

Understanding the mathematical foundations on which the neural networks and data prediction model works enables to reverse engineer the algorithm of machine learning model.

Chapter 4

Experiments and Results

4.1 Experimental Setup

The experimental setup includes the selection of appropriate datasets, splitting the data into training and testing sets, and tuning hyperparameters for each model.

```
J1 = df[df[’Junction’]==1]
J2 = df[df[’Junction’]==2]
J3 = df[df[’Junction’]==3]
J4 = df[df[’Junction’]==4]
vehciles_at_J1 = J1[’Vehicles’]
vehciles_at_J2 = J2[’Vehicles’]
vehciles_at_J3 = J3[’Vehicles’]
vehciles_at_J3 = J3[’Vehicles’]
```

Listing 4.1: Data set fetching using Pandas

```
#save
#logic take input for first 5 hours and
# predict 6th hour
# [[1],[2],[3],[4],[5]] [6]
# [[2],[3],[4],[5],[6]] [7]
# [[3],[4],[5],[6],[7]] [8]
def df_to_X_y(df, window_size=5):
df_as_np = df.to_numpy()
X =[]
y = []
for i in range(len(df_as_np)-window_size):
row = [[a] for a in
X.append(row)
label = df_as_np[i+window_size]
y.append(label)
return np.array(X), np.array(y)
```

Listing 4.2: Sample Result

4.2 Performance Metrics

Performance metrics used to evaluate the models include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).

```
model1 = Sequential()
model1.add(LSTM(64, input_shape=(WINDOW_SIZE, 1)))
model1.add(Dense(8, 'relu'))
model1.add(Dense(1))
model1.compile(loss=MeanSquaredError(), optimizer=Adam(learning_rate=0.001),
metrics=[RootMeanSquaredError()])
model1.summary()
```

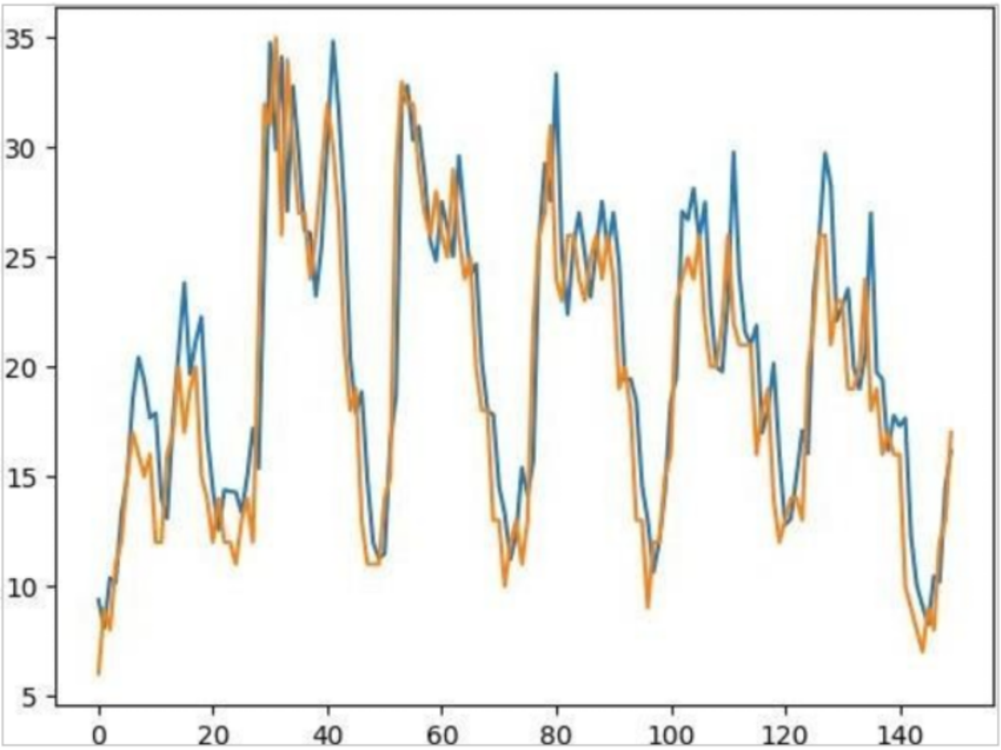
Listing 4.3: Performance Metrics

4.3 Results and Analysis

The results of the experiments are analyzed to compare the performance of different models and identify the most effective approaches for traffic flow prediction.

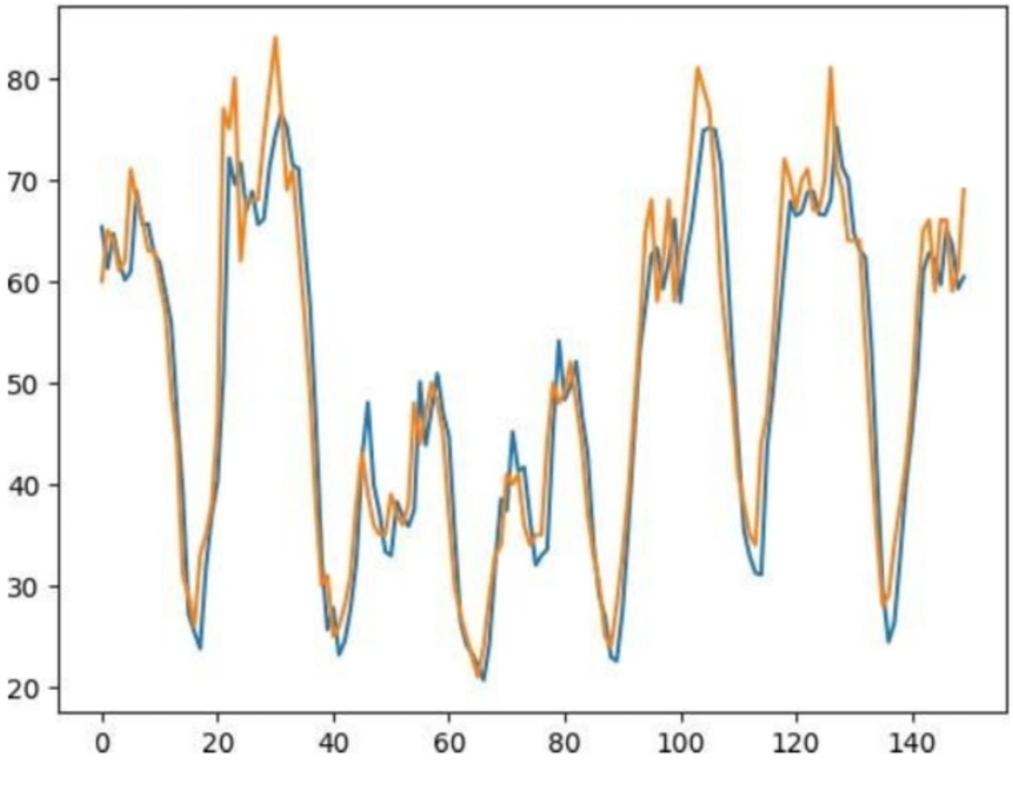
```
from tensorflow.keras.models import load_model
model1 = load_model('model1/')
train_results['Train Predictions'][:150].plot()
train_results['Train Actual'][:150].plot()
```

Listing 4.4: Performance Metrics



```
val_predictions = model1.predict(X_val).flatten()
val_results = pd.DataFrame(data={'Validation Predictions':val_predictions, '
Validation Actual':y_val})
val_results
val_results['Validation Predictions'][:150].plot()
val_results['Validation Actual'][:150].plot()
```

Listing 4.5: Performance Metrics


80
70
60
50
40
30
20
0
20
40
60
80
100
120
140

List of Figures

List of Tables

Listings

References

fedesoriano 2021.
Heart Failure Prediction Dataset.
Available at: `https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction`
Accessed on: 03/05/2024.

Goodfellow, Ian 2016.
NIPS 2016 Tutorial: Generative Adversarial Networks.
Available at: `https://arxiv.org/pdf/1701.00160.pdf`
Accessed on: 03/09/2023.