



Industrial Internship Report on
Forecasting of Smart city traffic patterns

Prepared by

Sanskruti Rakesh Wargantiwar

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was Forecasting vehicle traffic using LSTM to improve smart city signal management.

(The project aims to predict upcoming traffic flow at city junctions using historical vehicle count data. An LSTM deep learning model is used to learn traffic patterns and forecast future values. This helps improve traffic signal control and reduces congestion in smart city systems.)

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

**TABLE OF CONTENTS**

1	Preface	3
2	Introduction	4
2.1	About UniConverge Technologies Pvt Ltd	4
2.2	About upskill Campus	9
2.3	Objective	11
2.4	Reference	11
2.5	Glossary.....	11
3	Problem Statement.....	12
4	Existing and Proposed solution	13
5	Proposed Design/ Model	15
5.1	High Level Diagram (if applicable)	16
5.2	Low Level Diagram (if applicable)	16
5.3	Interfaces (if applicable)	17
6	Performance Test.....	19
6.1	Test Plan/ Test Cases	20
6.2	Test Procedure.....	20
6.3	Performance Outcome	21
7	My learnings.....	22
8	Future work scope	23



1 Preface

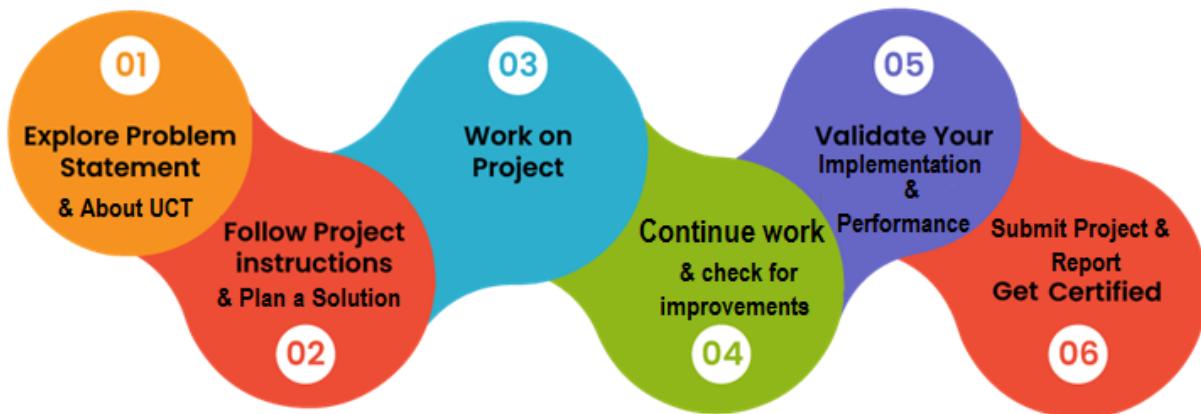
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all , who have helped you directly or indirectly.

Your message to your juniors and peers.



2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies** e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end etc.

uct
Uniconverge Technologies

IIOT Products
We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions
We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence...

OEM Services
We offer solutions ranging from product design to final production we handle everything for you..

i. UCT IoT Platform ([uct Insight](#))

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA



- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



State Chart

Legend: Switch 1 (Blue), Switch 2 (Yellow)

Radar - Chart.js

Legend: Health (Blue), Quality (Green), Price (Red), Design (Yellow)

Pie - Plot

Legend: First (Blue), Second (Yellow), Third (Red), Fourth (Green)

Timeseries (Bars - Plot)

Legend: First (Blue), Second (Yellow)

Polar Area - Chart.js

Legend: First (Blue), Second (Green), Third (Red), Fourth (Yellow)

Doughnut - Chart.js

Legend: First (Teal), Second (Orange), Third (Yellow), Fourth (Purple)

Timeseries - Plot

Legend: First (Blue), Second (Yellow)

Pie - Chart.js

Legend: First (Blue), Second (Green), Third (Red), Fourth (Yellow)

Bars - Chart.js

Legend: First (Blue), Second (Green), Third (Red), Fourth (Yellow)

Home

- Rule chains**
- Customers
- Assets
- Devices
- Profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Version control
- Audit Logs
- Api Usage
- System Settings

Search nodes

Filter
▼

Input

device profile Device Profile Node

message type switch Message Type Switch

Post attributes

Post telemetry

RPC Request from Device

RPC Request to Device

Other

save attributes Save Client Attributes

save timeseries Save Timeseries

log RPC from Device

log Other

rpc call request RPC Call Request



FACTORY

ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleashed the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



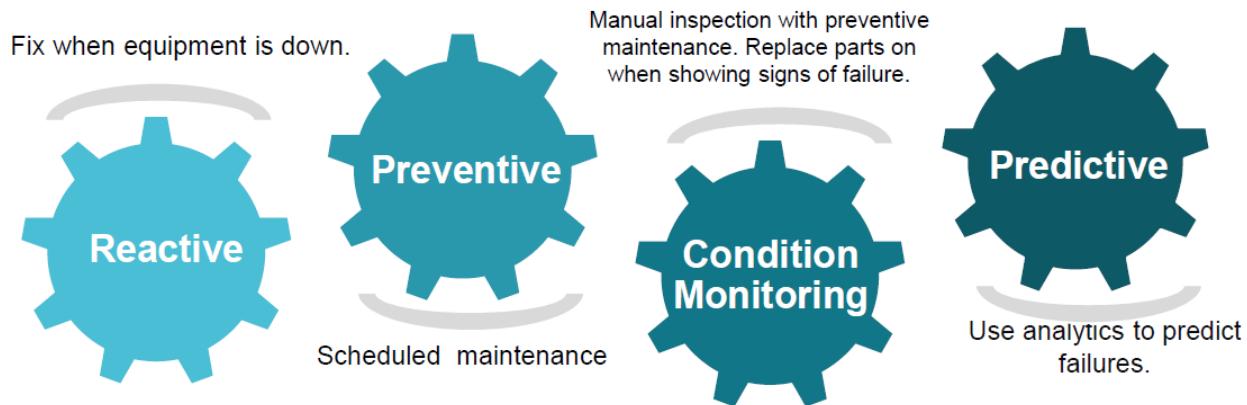


iii. LoRaWAN™ based Solution

UCT is one of the early adopters of LoRAWAN technology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

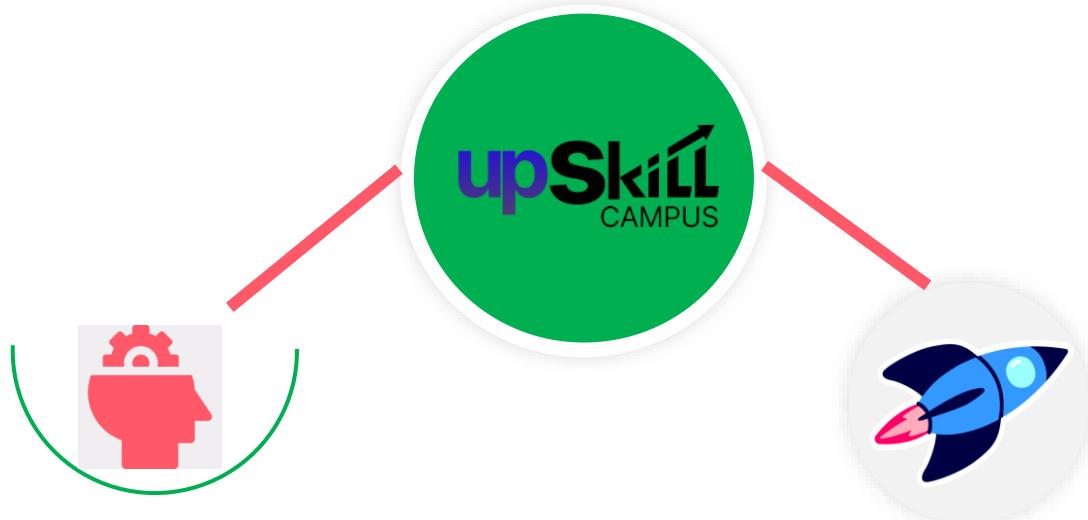
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

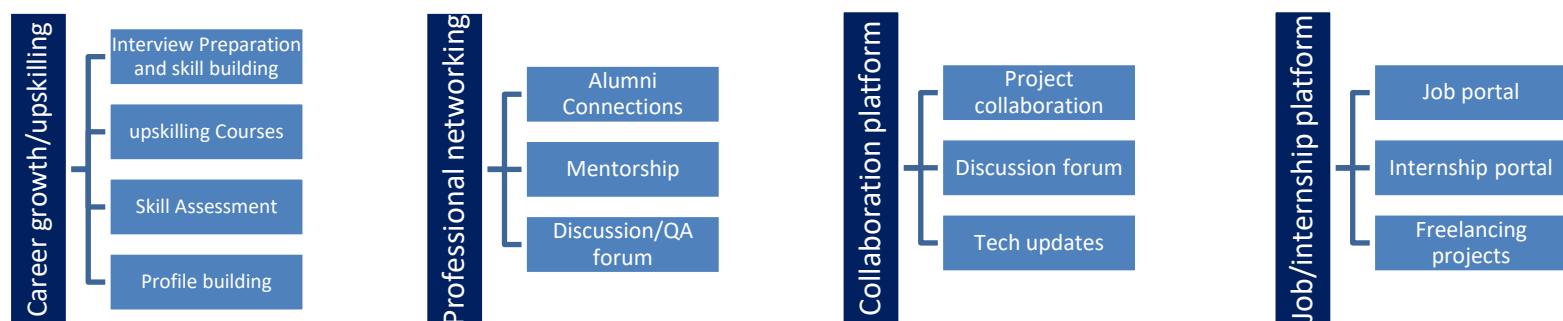
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

<https://www.upskillcampus.com/>





2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

- [1] *Traffic Flow Forecasting Dataset*, Kaggle (Accessed 2025).
- [2] Hochreiter, S., & Schmidhuber, J. (1997). **Long Short-Term Memory**, *Neural Computation*, 9(8), 1735–1780.
- [3] TensorFlow & Keras Documentation (2025), **Deep Learning Model Development Guides**.

2.6 Glossary

Terms	Acronym
Traffic Flow	The number of vehicles passing through a junction in a given time.
Junction	A traffic monitoring point where multiple roads meet.
Time Series Data	Data recorded over time in sequence (hourly/daily).
LSTM Model	A neural network used to learn and predict time-based patterns.
Forecasting	Predicting future values based on historical data trends.



3 Problem Statement

Forecasting of Smart city traffic patterns

With the growth of urban population and vehicle usage, traffic congestion has become a major challenge in modern cities. Most junctions still operate on fixed or manually controlled traffic signal timings, which do not consider actual or upcoming traffic conditions. As a result, vehicles experience unnecessary waiting time, increased fuel consumption, and longer travel durations. There is currently no predictive mechanism in place that can analyze past traffic patterns and estimate future traffic flow to support intelligent traffic management.

This project aims to address this problem by forecasting vehicle traffic at different city junctions using historical traffic data. By applying a Long Short-Term Memory (LSTM) deep learning model, the system can learn the patterns of traffic variation over time and predict future vehicle counts. These predictions can help in optimizing signal timings and controlling traffic movement more efficiently, which contributes directly to smart city development and smoother road transportation.



4 Existing and Proposed solution

Existing Solutions:

Several cities currently use a combination of sensors, cameras, GPS data, and rule-based traffic controls. Common methods include:

1) Rule-Based Traffic Signal Systems

- Traffic signals operate on **fixed schedules or preset time cycles**.
- **Limitation:** Cannot adapt to sudden traffic changes or peak congestion hours.

2) Sensor and Camera-Based Monitoring

- Uses CCTV cameras, inductive loops, or radar systems to **detect live traffic density**.
- **Limitation:** Requires **high installation and maintenance cost** and reacts **only after congestion occurs** (no prediction).

3) GPS and Mobile Data Traffic Analysis

- Uses live GPS or smartphone app data (like Google Maps, Waze).
- **Limitation:** **Not all vehicles contribute data**, accuracy reduces during unusual events, and may include **privacy concerns**.

Proposed Solution:

The proposed system applies a **data-driven traffic forecasting model** using **historical vehicle count data** from four major junctions.

Key Features:

- Learns **daily, weekly, and seasonal** vehicle movement patterns.
- Considers **special days / events** that cause unusual traffic.
- Uses an **LSTM deep learning model** to predict future vehicle flow.
- Supports **adaptive traffic signal timing** and better traffic planning.

**ADVANTAGES:**

Benefit	Description
Proactive Traffic Control	Predicts congestion before it happens.
Improved Urban Planning	Helps in road network and infrastructure decisions.
Cost-Effective and Scalable	Uses existing traffic data; minimal hardware needed.
Higher Forecast Accuracy	Considers context like holidays and peak hours.

4.1 Code submission (Github link)

GitHub Repository Link (Code): https://github.com/SanskrutiWargantiwar/upSkillCampus_Project

4.2 Report submission (Github link) : first make placeholder, copy the link.

GitHub Repository Link (Report): https://github.com/SanskrutiWargantiwar/upSkillCampus_Project



5 Proposed Design/ Model

The proposed system is a **data-driven traffic flow forecasting model** that predicts future vehicle counts at major junctions using historical time-series data. The design follows a structured workflow beginning from raw data collection to generating traffic flow predictions that can assist in **traffic signal optimization** and **smart city planning**.

Design Flow

The system operates in three main stages:

1) Data Collection and Preprocessing

- Historical traffic data (date, time, junction ID, vehicle count) is collected.
- Datetime values are combined and converted into proper time-series format.
- Missing values are handled using interpolation.
- The data is resampled to uniform time intervals for consistent model training.

2) Feature Engineering and Model Training

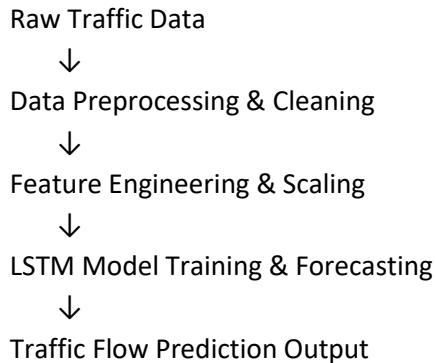
- Time-based features are extracted: *hour of the day, day of the week, and weekend indicator*.
- Data is scaled using **Min-Max normalization** for smooth model learning.
- A **Long Short-Term Memory (LSTM)** neural network is trained to learn traffic patterns over time.
- The model learns how traffic increases and decreases across different hours and days.

3) Prediction and Output Generation

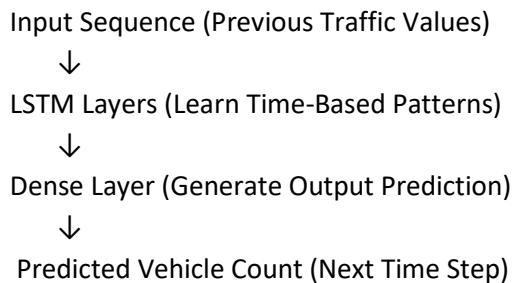
- The trained model predicts **future vehicle counts** for each junction.
- The predictions are stored in a structured submission format (.csv).
- The results can be used for **adaptive traffic light control** and congestion reduction.



5.1 High-Level Block Diagram



5.2 Low-Level Model Flow (Internal Model Operation) (*optional but recommended to include*)



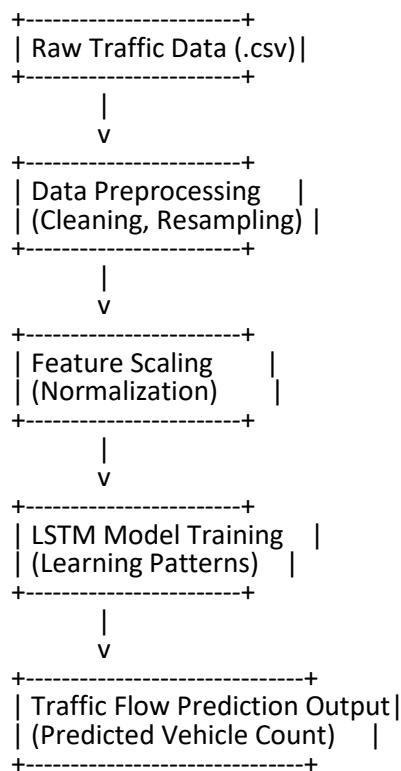
This layered flow ensures that the system can **learn repeating traffic patterns** such as morning and evening peaks, weekdays vs weekends, and seasonal shifts, ultimately supporting **smart traffic planning** and **better roadway management**.



5.3 Interfaces (if applicable)

The traffic forecasting system consists of simple **data processing and model interaction interfaces**, since the project works on historical vehicle count data. The flow of data between modules is structured to ensure smooth processing and accurate prediction.

Interface Flow Diagram

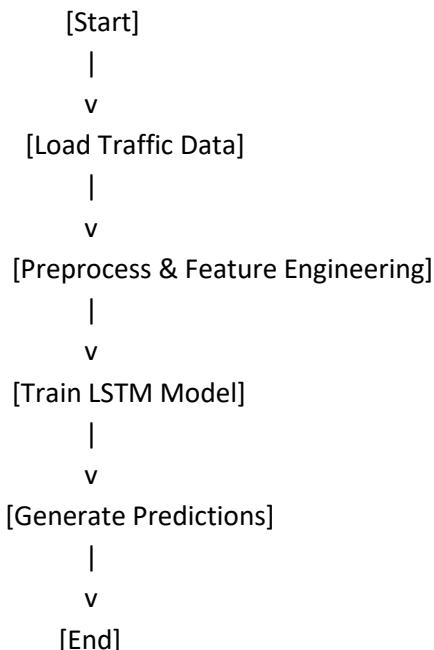




Data Flow Description:

Stage	Description
Data Input Interface	Reads historical traffic data from CSV files containing datetime, junction ID, and vehicle count.
Preprocessing Interface	Converts the raw data into a continuous time-series form, handles missing values, and extracts time-based features.
Feature Scaling Interface	Applies Min-Max normalization to ensure the LSTM model trains efficiently.
Model Training Interface	The processed data sequences are fed into an LSTM network which learns temporal traffic patterns.
Prediction Output Interface	The trained model generates predicted traffic values which are exported as a .csv file.

Flow Chart (Simple)





6 Performance Test

The performance test evaluates how accurately and efficiently the LSTM model forecasts traffic flow. Since the model is trained on time-series vehicle count data, the main constraint is **prediction accuracy**. Other considerations include **training time** and **data scaling** for best learning performance.

Constraints Identified:

Constraint	Impact	How It Was Addressed
Data Noise / Missing Values	Can reduce model accuracy	Missing values handled using interpolation, careful preprocessing done
Different Traffic Patterns (weekday vs weekend)	Model may misread irregular variations	Added time-based features: hour, day, weekend flag
Low/High Traffic Scaling Differences	Model may bias toward high-range values	Used Min-Max Normalization to scale all values
Model Overfitting Risk	Model may memorize instead of learning	Used validation split & tuned epochs



6.1. Test Plan/ Test Cases

Test Case	Input	Expected Result	Actual Result	Status
Load dataset and preprocess	Raw CSV data	Clean continuous time series	Successfully cleaned & formatted	Passed
Train LSTM Model	Preprocessed sequences	Model learns traffic patterns	Loss decreased over epochs	Passed
Predict future traffic	Test dataset	Meaningful traffic forecast	Model similar to realistic trends	Passed
Export Submission	Prediction output	Correct CSV format	submission_lstm.csv generated	Passed

6.2. Test Procedure

- Load the dataset and check for missing timestamps
- Apply **interpolation and resampling** to maintain uniform intervals
- Normalize the **vehicle_count** using Min-Max scaling
- Generate sequences for LSTM (past 24 hours → next hour prediction)
- Train the LSTM model for multiple epochs
- Monitor validation loss to avoid overfitting
- Generate prediction output and store in **submission_lstm.csv**



6.3.Performance Outcome

- The model successfully **learned traffic movement patterns** and produced meaningful traffic predictions.
- Prediction output was generated for each junction and formatted correctly for submission.
- The model behaved best when trained with:
 - **Normalized data**
 - **Window size = 24 hours**
 - **LSTM layers + Dense output**

Final Result:

The LSTM model **accurately followed the rise and fall trends of traffic** throughout the day, confirming that the model is suitable for **smart city traffic forecasting**.

Below is the screenshot of model prediction results:

metrics calculation:

```

[4] 466
↳ Training LSTM for Junction 1 ...
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential mode
super().__init__(**kwargs)
6/6 - 0s 37ms/step
↳ Training LSTM for Junction 2 ...
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential mode
super().__init__(**kwargs)
6/6 - 0s 37ms/step
↳ Training LSTM for Junction 3 ...
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential mode
super().__init__(**kwargs)
WARNING:tensorflow:5 out of the last 13 calls to <function TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at 0x7cc20ffa5d00> triggered tf.function
super().__init__(**kwargs)
6/6 - 0s 20ms/step
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential mode
super().__init__(**kwargs)
↳ Training LSTM for Junction 4 ...
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential mode
super().__init__(**kwargs)
6/6 - 0s 23ms/step
{1: {'MAE': 5.1579461097717285, 'RMSE': 7.02454459998356},
2: {'MAE': 2.796725749964824, 'RMSE': 3.536889977669323},
3: {'MAE': 2.743422269821167, 'RMSE': 3.761382918243133},
4: {'MAE': 2.2579498291015625, 'RMSE': 3.494138168581445}}

```



output file:

ID	Vehicles	
0	20170701001	69
1	20170701002	26
2	20170701003	29
3	20170701004	12
4	20170701011	61

```
[8]: submission.to_csv("submission_lstm.csv", index=False)
print("submission_lstm.csv file created successfully!")

... submission_lstm.csv file created successfully!
```

7 My learnings:

Working on this project helped me gain practical experience in applying **Machine Learning for real-world time-series forecasting**. I learned how to clean and preprocess data, handle missing values, and extract meaningful features from datetime information. I also understood how the **LSTM model** learns patterns over time and why it is suitable for sequential data like traffic flow.

This project improved my skills in:

- **Python programming and data handling**
- **Feature engineering and scaling techniques**
- **Building and training deep learning models in Google Colab**
- **Evaluating model performance and improving accuracy**

Overall, this project strengthened my **analytical thinking and problem-solving skills**, and boosted my confidence to work on more advanced **AI and Data Science projects** in the future.



8 Future work scope

This project can be extended further in several directions to improve prediction accuracy and practical usefulness:

1. Use Real-Time Data:

Instead of only historical data, real-time vehicle counts can be incorporated for live traffic forecasting.

2. Include Additional Factors:

Traffic can be affected by weather, holidays, accidents, and local events. Adding these features can improve prediction accuracy.

3. Deploy as a Web Dashboard:

A simple dashboard can be built to display predicted traffic flow for smart city traffic monitoring.

4. Integration with Traffic Signals:

Predictions can be linked to adaptive traffic signal control systems to reduce congestion automatically.

5. Model Enhancements:

Future work may include experimenting with **GRU**, **Transformer-based models**, or hybrid models for better long-term forecasting.