SDAIA
الهيئة السعودية للبيانات والذكاء الاصطناعي
Saudi Data & AI Authority

أكاديمية طويق
Tuwaiq Academy

Traffic Flow Prediction

Mini Project T5 BootCamp

Team Work

Joud Tarek

Retaj Alanzii

Dana Almistdadi

Rana al-Harsan

Table of Contact

Traffic Prediction Overview

Objective

The idea for the project is to create a deep learning model for predicting traffic jams on roads by utilizing the data that is currently available, such as the time of day, the quantity and kind of cars on the road, and the type of traffic. Based on the given data, the model will forecast the kind and timing of congestion.

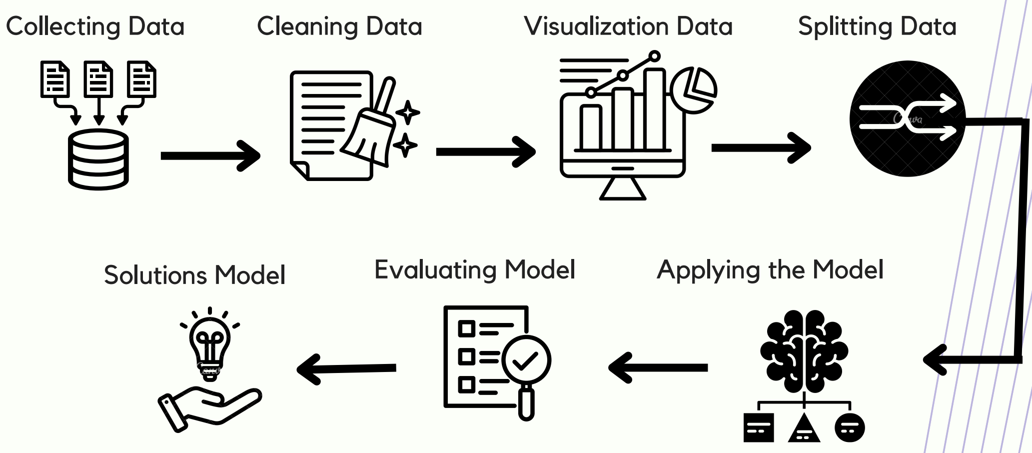
Data Source

The data was gathered from Kaggle data source.
The two data files that make up these are called "Traffic" and "Traffic in Two Months. "and To complete the dataset.

Source : Kaggle

Methodology

We did exploratory data analysis (EDA) after importing the dataset. We followed by carrying out data preprocessing, which included scaling and separating the data. Next, we used LSTM to create a Time Series model, tested it, and stored the data.



EDA & Visualization

Understanding Data

The dataset focuses on traffic congestion in urban areas. This dataset offers valuable insights for transportation planning, infrastructure development, and congestion management, enabling stakeholders to make informed decisions that enhance urban mobility and contribute to sustainable city planning. Updated every 15 minutes over one month, the dataset provides a comprehensive view of traffic patterns , and To complete the dataset, we also produced synthetic data .
It includes a traffic situation column with four categories: Heavy,High, Normal, and Low, which helps assess congestion severity.

Column Name	Description
Latitude :float	Longitude measures how far a location is east or west of the Prime
Longitude:float	Latitude measures how far a location is north or south of the Equator.
Day of the week : object	the day of the congestion occur
CarCount : int	Count how many car occur in the Traffic
BikeCount : int	Count how many bike occur in the Traffic
BusCount : int	Count how many bus occur in the Traffic
TruckCount : int	Count how many Truck occur in the Traffic
Total : int	Count Total Cars occur in the Traffic
Traffic Situation : Object	Describe the Traffic situation if it's normal or havey ..etc

EDA & Visualization

Import libraries

in this first step of creating model , we applied libraries from tensorflow.keras to be able to create LSTM model.

```
[ ]  import numpy as np
     import pandas as pd
     import warnings
     warnings.filterwarnings("ignore")

     # For processing
     import math
     import random
     import datetime as dt
     import matplotlib.dates as mdates

     # For visualization
     import matplotlib.pyplot as plt
     from mplfinance.original_flavor import candlestick_ohlc
     from sklearn.preprocessing import LabelEncoder
     import seaborn as sns
     # Libraries for model training
     from sklearn.preprocessing import MinMaxScaler
     from keras.models import Sequential
     from keras.layers import SimpleRNN, Dense, Dropout
     from keras.callbacks import ModelCheckpoint, EarlyStopping
     from sklearn.metrics import mean_squared_error
     from sklearn.model_selection import train_test_split
     from imblearn.over_sampling import RandomOverSampler
     from tensorflow.keras.layers import LSTM ,Dense,Dropout
     from sklearn.metrics import classification_report,confusion_matrix
```

Load Dataset

In the loading dataset step, we combined multiple datasets and also added synthetic data to enrich the overall dataset.

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')

     Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ]  traffic_df=pd.read_csv("/content/drive/MyDrive/T5 project /Dataset/Expanded_Traffic_With_Coords.csv",parse_dates=['DateTime'], index_col='DateTime')
     traffic_df.head()
```

EDA :

Step 3: Data Preparation and Preliminary Analysis

Initial Dataset Examination

We commenced our analysis with the Traffic_csv dataset. The following steps were undertaken to understand and prepare the data:

Previewing the Dataset:

The first and last five rows of the dataset were examined to gain an overview of the data structure and the types of values present in each column.

This preliminary inspection helped in identifying the nature of the data and provided initial insights into the distribution of traffic-related features.

Data Type Analysis:

The data types of each column were reviewed using the info() method. Understanding the data types was crucial for identifying categorical and numerical features, as well as for determining any necessary conversions or adjustments.

Null Value Detection:

A thorough check for null values was performed to identify any missing data. This step ensured that potential gaps in the dataset were recognized early, allowing for appropriate handling strategies to be implemented.

Duplicate Data Check:

The dataset was examined for duplicate entries. Ensuring data uniqueness was vital for maintaining the integrity and reliability of the analysis.

These steps were repeated for the second dataset, TrafficTwoMonth_csv, to ensure that both datasets were thoroughly examined and prepared for subsequent analysis.

EDA :

Data Visualization:

To deepen our understanding of the datasets, various visualizations were created. These visualizations helped in exploring relationships between features, identifying patterns, and gaining insights into the overall distribution of the data. This exploratory analysis was crucial for informing the next stages of the project.

Dataset Merging and Preparation for Modeling

Prior to developing the Neural Network model, the two datasets were merged to create a comprehensive dataset.

The following preparatory steps were then executed:

Label Encoding:

Categorical features were converted into numerical values through label encoding. This transformation was necessary to enable the application of machine learning algorithms that require numerical input.

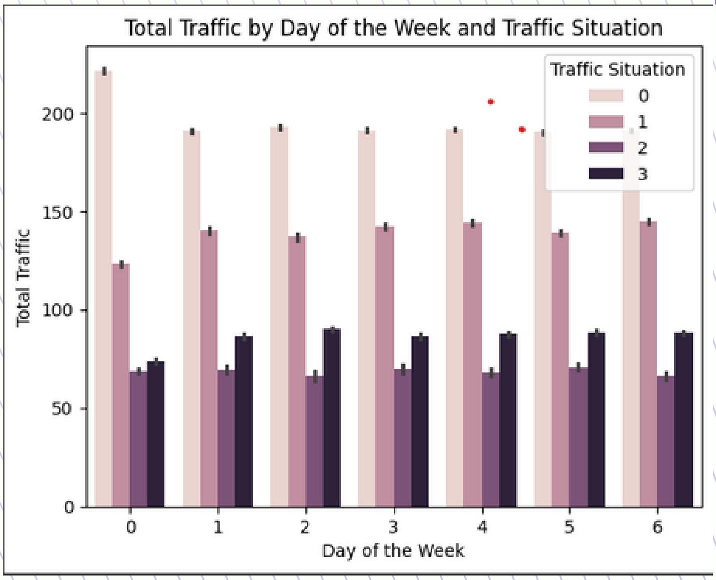
MinMaxScaler:

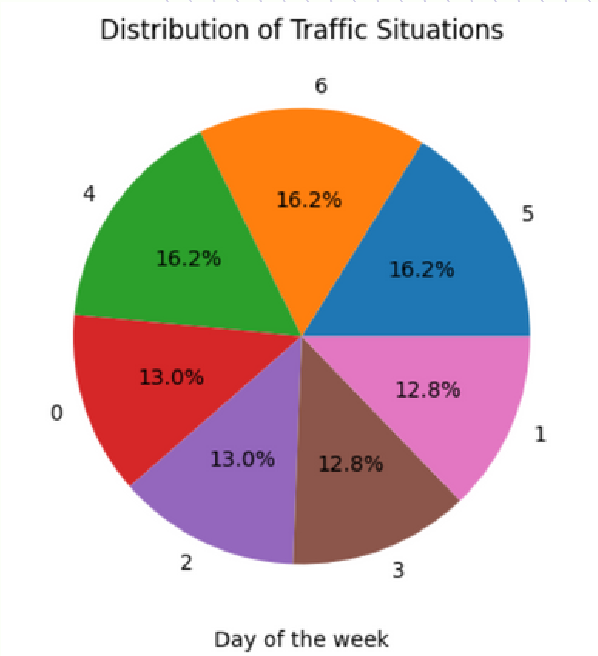
The dataset was normalized using MinMaxScaler, ensuring that all features were scaled consistently within a specific range. This step was essential for optimizing the performance of the Time Series model, specifically the LSTM, by preventing any single feature from disproportionately influencing the results. By carefully executing these steps, the dataset was cleaned, analyzed, and fully prepared for the development of the LSTM model

visualization the dataset:

Bar Plot:

applying Seaborn to create a bar plot that classifies traffic based on its condition and displays total traffic by day of the week.





Pie Chart:

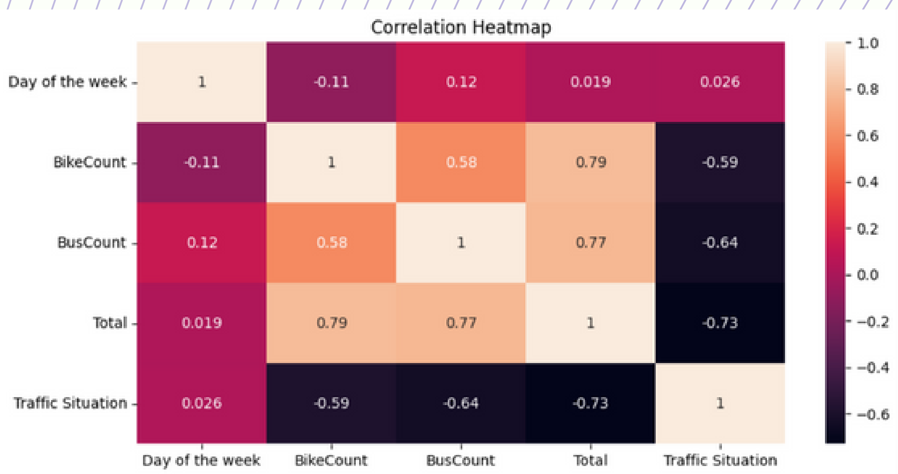
Traffic Distribution by Day of the Week

The pie chart below illustrates the distribution of traffic data across different days of the week. Each segment of the pie represents the proportion of traffic recorded on a specific day.

Heatmap:

Correlation of Numerical Data

The heatmap below visualizes the correlation matrix of the numerical features in the dataset. After removing the Date, TruckCount, and CarCount columns, the remaining numerical columns are analyzed to identify how closely they are related to each other.



LSTM Model

For time series forecasting and other sequential tasks, the Long Short-Term Memory (LSTM) model is a form of recurrent neural network (RNN) that is specifically built to process and predict sequences of data.

LSTM is an advanced type of recurrent neural network (RNN) designed to remember information over long periods and handle sequential data that requires this capability. This makes it one of the powerful tools in time series analysis and sequential data processing.

```
[ ]   model=Sequential()
      model.add(LSTM(64, return_sequences=True, input_shape=(time_step,1)))
      model.add(LSTM(32, return_sequences=True, input_shape=(time_step,1)))
      model.add(LSTM(16, return_sequences=False))
      model.add(Dense(1))
      # model.add(Dropout(0.2))
      model.compile(loss='mean_squared_error', optimizer='adam')
```

```
model.summary()
```

Model: "sequential"

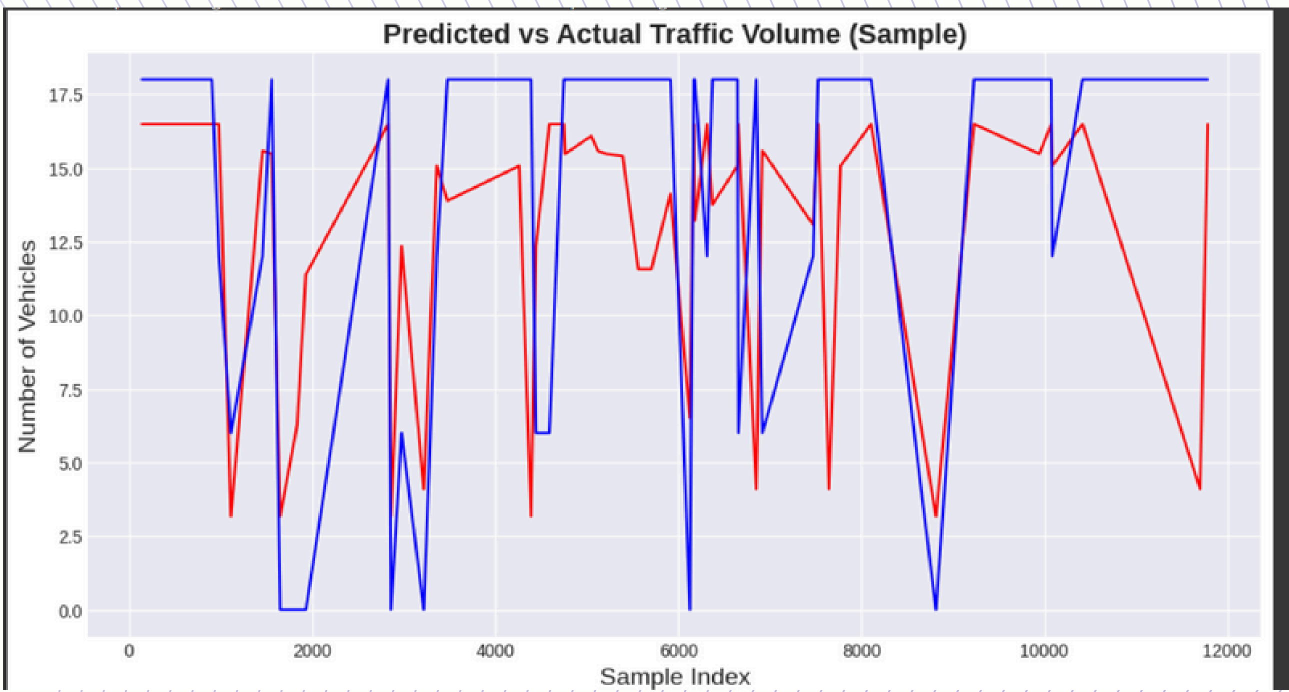
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 3, 64)	16,896
lstm_1 (LSTM)	(None, 3, 32)	12,416
lstm_2 (LSTM)	(None, 16)	3,136
dense (Dense)	(None, 1)	17

Total params: 32,465 (126.82 KB)
Trainable params: 32,465 (126.82 KB)
Non-trainable params: 0 (0.00 B)

LSTM Model Evaluation

The image displays the actual results and predictions of the LSTM model for predicting traffic volume. The red line represents the recorded vehicle numbers, while the blue line represents the predicted values. The graph shows a match between the actual and predicted values, suggesting the model's ability to predict patterns.

However, there are noticeable gaps or differences, suggesting the model struggles to capture complex patterns or noisy data. Significant differences between the actual and predicted values suggest the model might struggle with complex patterns or anomalies. If these differences persist, retraining the model with more data or tuning parameters could improve the model's performance.



Solutions

Smart Traffic Management:
Real-time traffic signal timing adjustments can be made by applying prediction results.

Emergency Management:
Using predictions, direct emergency vehicles (fire trucks, ambulances) along less crowded routes.

Dedicated Routes for Public Transportation:
Assign particular lanes for public transportation according to predictions.

Conclusion

In conclusion, the deep learning-based traffic congestion prediction study successfully showed how advanced algorithms can evaluate complicated information to predict traffic patterns. This project offers important insights that can optimize urban design, improve traffic management, and raise overall road safety by accurately predicting congestion. This creative method is a big step toward building more intelligent, effective transportation systems.