Smart Flow - AI-Based Traffic Light Management System

1. Introduction & Problem Statement

Real-World Problem

Urban traffic congestion is a critical issue, leading to increased travel times, fuel consumption, and environmental pollution. Traditional traffic light systems operate on fixed intervals, failing to adapt to real-time traffic fluctuations. This inefficiency results in unnecessary waiting times and imbalanced traffic flow across lanes.

Importance of an AI-Based System

An intelligent traffic light management system can:

- Dynamically adjust signal durations based on real-time and historical traffic density.
- Optimize green light durations to minimize overall waiting time.
- Improve traffic efficiency by predicting congestion patterns.
- Adapt to various traffic conditions, including peak hours and low-traffic periods.

2. Dataset & Preprocessing

Dataset Overview

The dataset used for training includes:

- **Time of Day:** Collected at hourly intervals over a two-year period.
- **Vehicle Density:** Normalized traffic density for each lane.
- **Current Green Duration:** Duration of the green signal for the active direction.
- **Traffic Volume Trends:** Historical traffic patterns for predictive modeling.

Data Preprocessing

To enhance model accuracy, the following preprocessing steps were applied:

- **Feature Engineering:**

The `time_of_day` feature was transformed using sine and cosine functions to capture its cyclic nature:

```
df['hour_sin'] = np.sin(2 * np.pi * df['hour'] / 24)
df['hour_cos'] = np.cos(2 * np.pi * df['hour'] / 24)
```

- **Normalization:**

Vehicle density values were normalized between 0 and 1 using `MinMaxScaler`.

MinMaxScaler().fit_transform(density_columns)

- **Target Variable Calculation:**
  - The `optimal_green_duration` was estimated using a weighted sum of past traffic densities.

The dataset was split into **80% training, 10% validation, and 10% testing**, ensuring a time-aware split to prevent data leakage.

3. Model Implementation & Evaluation

Model Selection

Since traffic patterns have temporal dependencies, a **Long Short-Term Memory (LSTM) network** was chosen to learn short-term and long-term fluctuations.

Model Architecture

- **Input Features:** 7 features (time features, vehicle densities, current green duration)
- **LSTM Layers:**
  - `LSTM(32, return_sequences=True)`
  - `LSTM(32, return_sequences=False)`
- **Fully Connected Layers:**
  - `Dense(16, activation='relu')`
  - `Dense(1)` (Output layer predicting the next optimal green duration)

Training Configuration

- **Optimizer:** Adam (learning_rate=0.0005, decay=1e-6)
- **Loss Function:** Mean Squared Error (MSE)
- **Batch Size:** 16
- **Epochs:** 20 (with early stopping to prevent overfitting)

4. Performance Evaluation

Key Metrics

Dataset	MSE	RMSE	MAE	R^2 Score

Train	18.74	4.33	3.11	0.9492
Validation	25.16	5.02	3.85	0.9364
Test	41.48	6.44	4.62	0.9401

Observations

- The model effectively adjusts green light durations based on traffic density variations.
- During **peak hours (e.g., 8 AM - 9 AM)**, the model assigns longer green durations to high-density lanes.
- At **low traffic periods**, it ensures fair signal distribution to avoid unnecessary delays.

5. Challenges & Future Improvements

Current Challenges

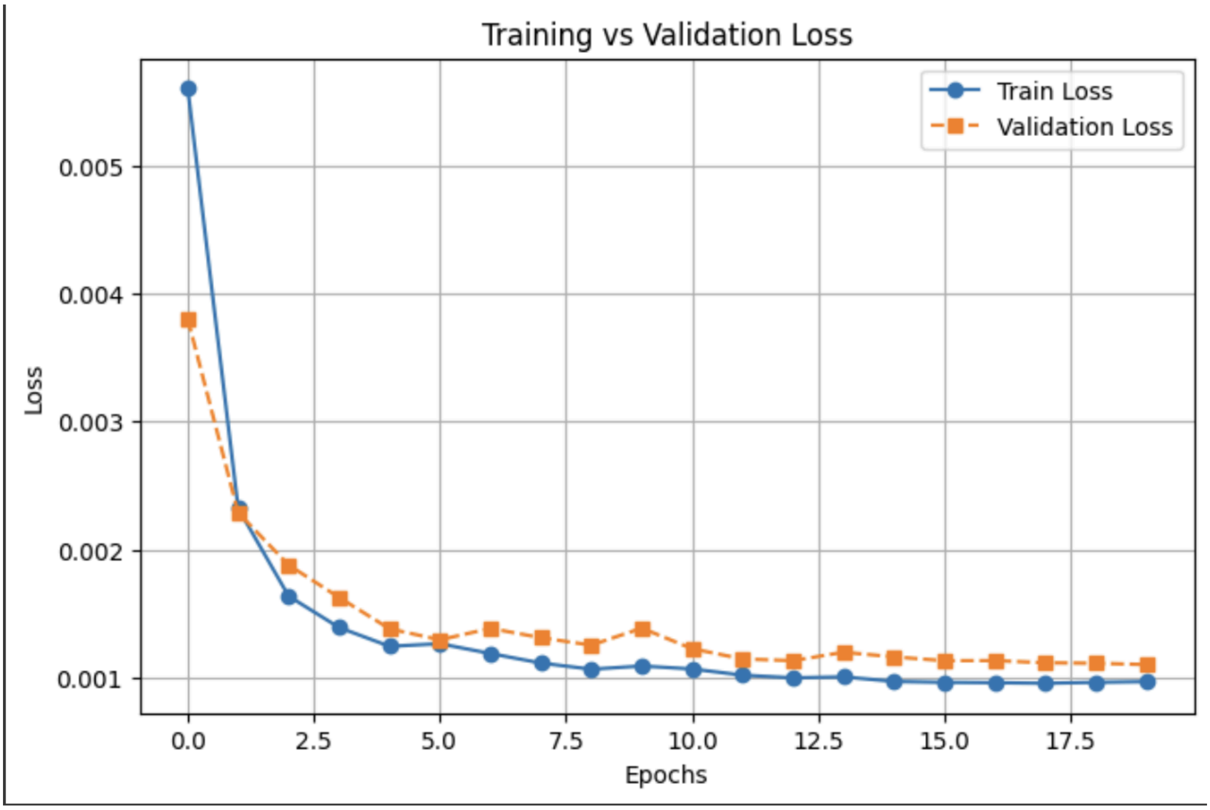
- **Real-World Data Collection Limitations:** Sensor inconsistencies require additional filtering.
- **Model Generalization:** Overfitting needs to be minimized by introducing dropout layers.
- **Hardware Constraints:** Real-time inference needs optimization for <100ms response time.

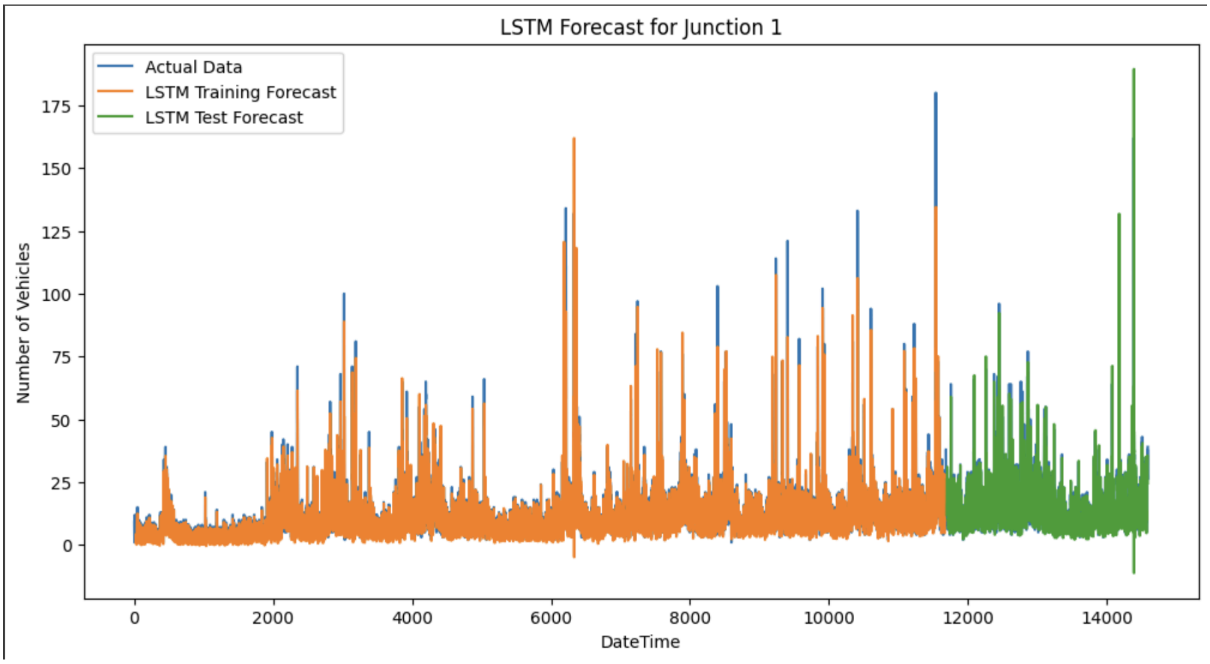
Future Enhancements

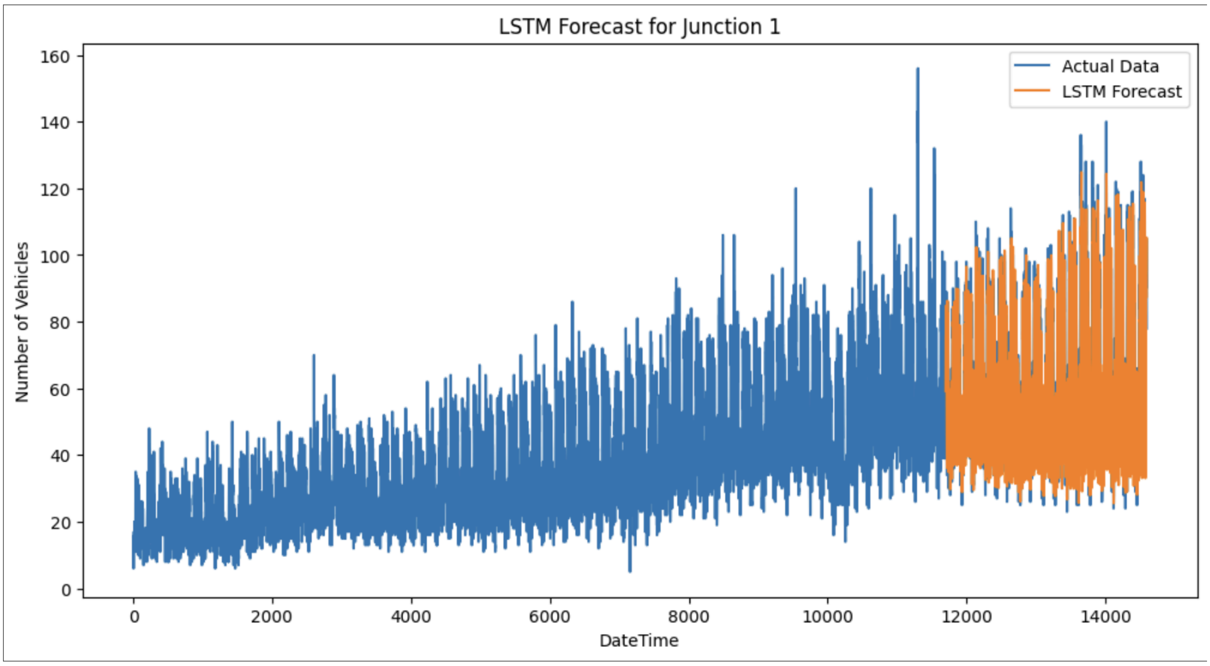
- **Reinforcement Learning Integration:**
    - Implement **Deep Q-Learning** to dynamically adjust signals based on real-time rewards (e.g., minimizing total wait time).
- **Emergency Vehicle Handling:**
    - Detect emergency vehicles and adjust signals to provide priority clearance.
- **Multi-Intersection Coordination:**
    - Expand the system to synchronize multiple intersections for city-wide traffic flow optimization.
- **Edge Deployment:**
    - Optimize the model for deployment on **low-power embedded devices (e.g., Raspberry Pi).**

6. Conclusion

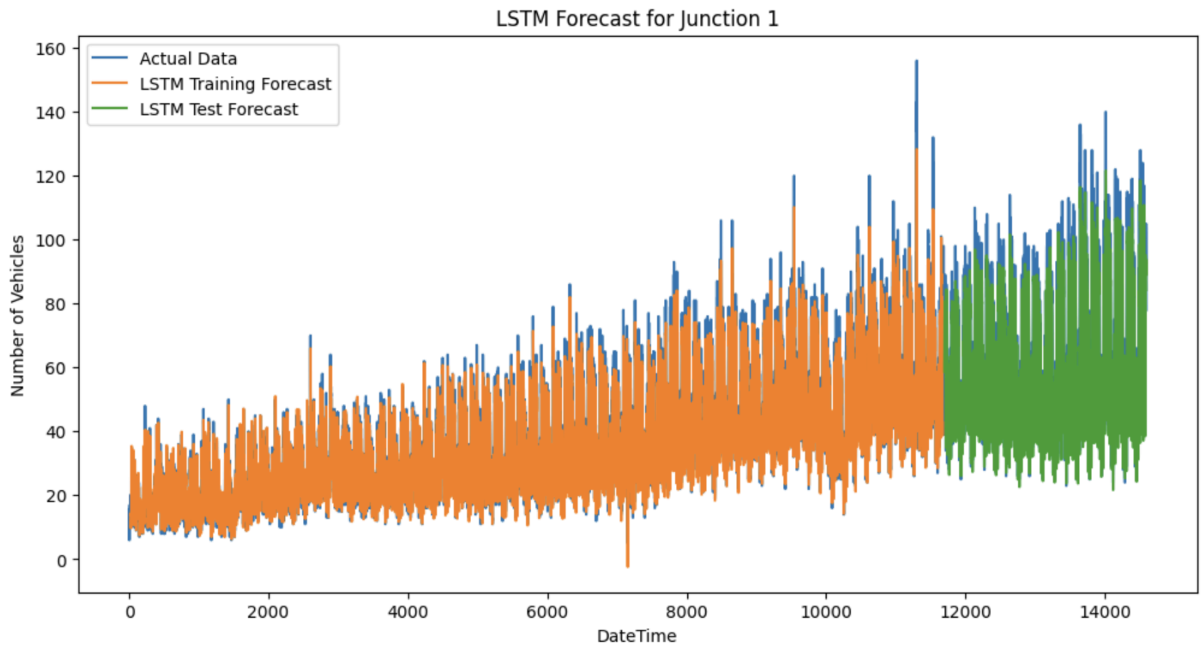
This project demonstrates the feasibility of an **AI-powered traffic light system** that dynamically adjusts green durations based on real-time and historical traffic density. The model effectively reduces waiting times, enhances traffic flow, and can be improved further using **reinforcement learning and multi-intersection coordination**. With real-world deployment, this system can contribute significantly to **smart city traffic optimization**.







LSTM Forecast for Junction 1
Actual Data
LSTM Forecast
Number of Vehicles
DateTime



LSTM Forecast for Junction 1
Actual Data
LSTM Training Forecast
LSTM Test Forecast
Number of Vehicles
DateTime