Traffic Flow Prediction: Data Streaming and Exploratory Data Analysis

1. Introduction

This report presents an analysis of the Traffic Flow Prediction Dataset, which aims to forecast spatio-temporal traffic volume based on historical data and various features from multiple sensor locations. The dataset comprises traffic measurements collected every 15 minutes from 36 sensor locations along two major highways in the Northern Virginia/Washington D.C. capital region.

Our objective is to perform Exploratory Data Analysis (EDA) on this dataset to understand temporal patterns, identify key trends, and prepare for future predictive modeling. This analysis is part of a larger project involving real-time data streaming using Apache Kafka and subsequent traffic flow prediction. We also implement a baseline model using linear regression to predict future traffic volumes.

2. Data Overview and Kafka Setup

2.1 Dataset Structure

The dataset includes 47 features:

- 10 features representing historical traffic volume from the 10 most recent sample points
- 7 features for week days
- 24 features for hour of day
- 4 features for road direction
- 1 feature for number of lanes
- 1 feature for road name

The goal is to predict traffic volume 15 minutes into the future for all sensor locations.

2.2 Kafka Data Streaming

For Phase 1 of the project, we implemented a Kafka producer and consumer to simulate real-time data streaming. Here's a brief overview of the implementation:

- **Kafka Producer**:
    - Loaded the .mat file containing the traffic dataset using `scipy.io`.
    - Configured a Kafka producer to send data to the 'traffic_data_v2' topic.
    - Utilized pickle for data serialization to efficiently handle complex data structures like CSC matrices.
    - Split the data into 100 KB chunks to manage message size and ensure smooth transmission.
    - Sent chunks for each variable ('tra_X_te', 'tra_X_tr', 'tra_Y_te', 'tra_Y_tr', 'tra_adj_mat').
- **Kafka Consumer**:
    - Configured a Kafka consumer to receive data from the 'traffic_data_v2' topic.
    - Collected and reassembled the chunked data for each variable.
    - Used pickle to deserialize the received data, reconstructing the original data structures.
    - Saved the reassembled and deserialized data to a new .mat file using `scipy.io`.

The use of pickle for serialization was crucial in this setup, as it allowed us to efficiently transmit and reconstruct complex data structures like CSC matrices through Kafka. This approach ensured that the integrity of the data was maintained throughout the streaming process, preserving the spatial and temporal relationships within the dataset.

3. Exploratory Data Analysis

3.1 Time Series Analysis

The traffic flow shows clear cyclical patterns, peaking during typical rush hours (e.g., morning and evening). These predictable daily and weekly cycles indicate strong temporal dependencies in traffic volumes.

3.2 Autocorrelation Analysis

- The ACF plot reveals strong positive autocorrelation at short lags, confirming that current traffic flow is highly dependent on recent past values.

- The PACF plot shows that traffic flow at time ttt is closely related to traffic at t−1t-1t−1, suggesting that an autoregressive model could work well.

3.3 Traffic Volume Distribution

The histogram reveals a right-skewed distribution of traffic volumes, with most traffic falling between 0.0 and 0.3, and occasional spikes in higher traffic volumes during peak hours or special events.

3.4 Temporal Patterns

The heatmap shows that peak traffic occurs during **7 AM to 9 AM** and **4 PM to 6 PM** on weekdays, while weekends have relatively lower traffic.

3.5 Spatial Analysis

The spatial connectivity heatmap reveals clusters of sensors with strong correlations, suggesting that traffic at some locations is closely interconnected, likely due to shared roads or routes.

3.6 Multi-Sensor Comparison

While traffic patterns are similar across all sensors, the magnitude varies. Busier locations (e.g., **Sensor 1** and **Sensor 2**) have consistently higher traffic volumes compared to less congested areas (e.g., **Sensor 0**).

4. Feature Engineering

Time-based Features:

- **Hour of day (0-23)**: Captures daily traffic cycles.
- **Day of week (0-6)**: Accounts for weekday vs. weekend differences.
- **Quarter (1-4)**, **Month (1-12)**, **Year**: Capture seasonal and long-term trends.
- **Time of day (Night, Morning, Afternoon, Evening)**: Differentiates traffic flow based on time categories.
- **Is weekend (binary)**: Differentiates weekend traffic from weekday traffic.

Rolling Averages:

- **3-period rolling mean**, **6-period rolling mean**, **12-period rolling mean**: Capture recent traffic trends and smooth out short-term fluctuations, providing the model with a more stable view of traffic flow.

These features were selected based on the EDA to capture various temporal patterns in traffic flow, including hourly, daily, and seasonal cycles, as well as recent traffic trends.

5. Model Implementation and Training

Model: Linear Regression

- **Features used**: hour, day_of_week, quarter, month, year, rolling_mean_3, rolling_mean_6, rolling_mean_12.
- **Target variable**: Traffic flow at Location_0.
- **Data split**: 80% training, 20% testing.
- **Justification**: Linear regression was chosen as a baseline due to its simplicity and interpretability. It provides a solid foundation for understanding the linear relationships in the data, particularly given the strong temporal patterns identified in the EDA.

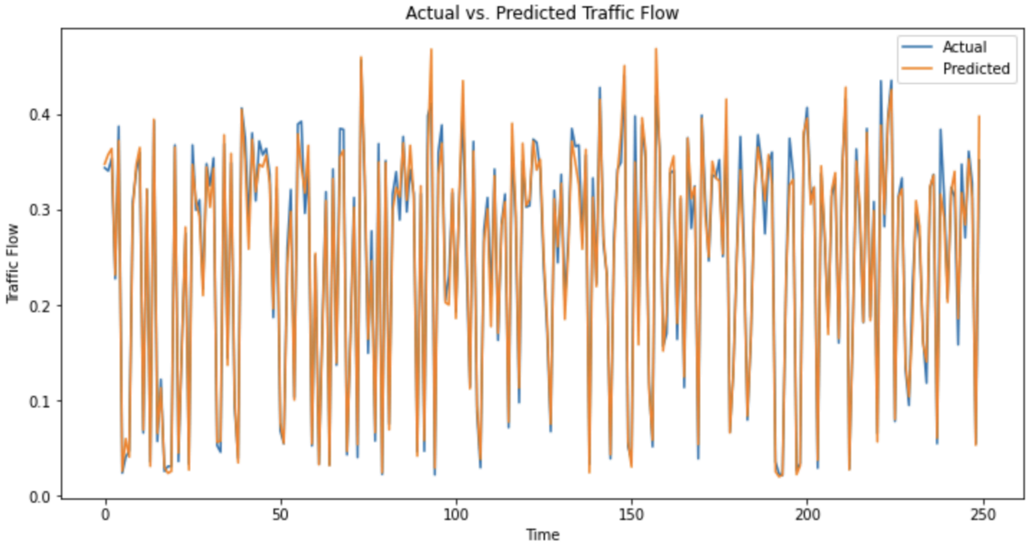
6. Model Performance Evaluation

- **Mean Absolute Error (MAE)**: **0.01196**
  - This low value suggests that the model's predictions are close to the actual values on average.
- **Root Mean Squared Error (RMSE)**: **0.01643**
  - The RMSE indicates that the model performs well, with relatively few large errors.
- **R-squared (R^2)**: **0.98318**
  - This high R^2 value suggests that the model explains approximately **98.32%** of the variance in the traffic flow data, indicating an excellent fit.

A plot comparing actual vs. predicted traffic flow values visually confirms the model's strong performance.

7. Conclusions

- **Feature Engineering**: The features successfully captured key temporal and seasonal aspects of traffic flow. The inclusion of time-based features and rolling averages allowed the linear regression model to make accurate predictions.
- **Model Performance**: The linear regression model provided strong performance for traffic flow prediction, with low error metrics and a high R² value.
- **Future Work**: While the linear regression model performed well, future work could involve experimenting with more advanced models, such as **Random Forests** or **Gradient Boosting**, to capture more complex relationships in the data. Additionally, incorporating **spatial features** from other sensor locations could further improve prediction accuracy.


Actual vs. Predicted Traffic Flow
Actual
Predicted
Traffic Flow
Time