Report on Model Evaluation and Refinement

1. Project overview:

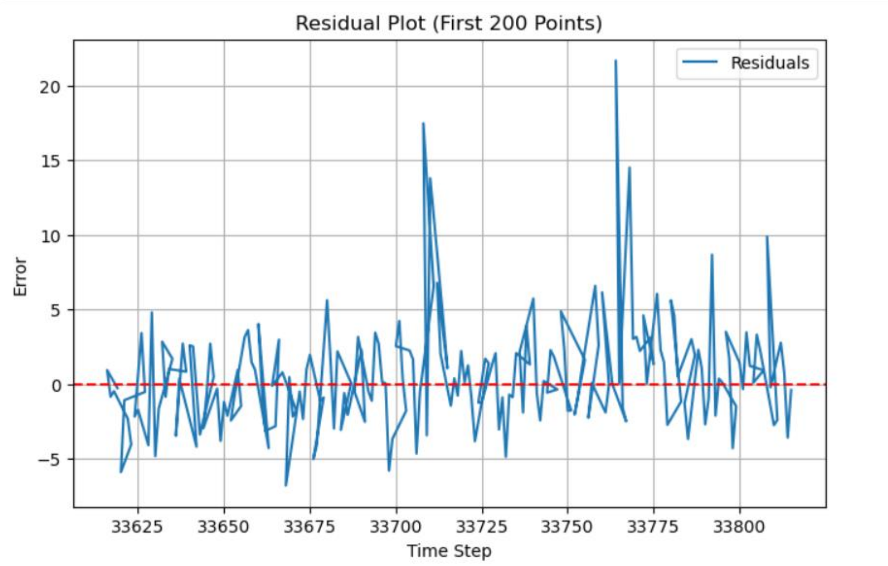
- Develop a predictive model to accurately forecast hourly vehicle volumes at urban road junctions based on historical, weather, and event-related data.

2. Model Development and Training:

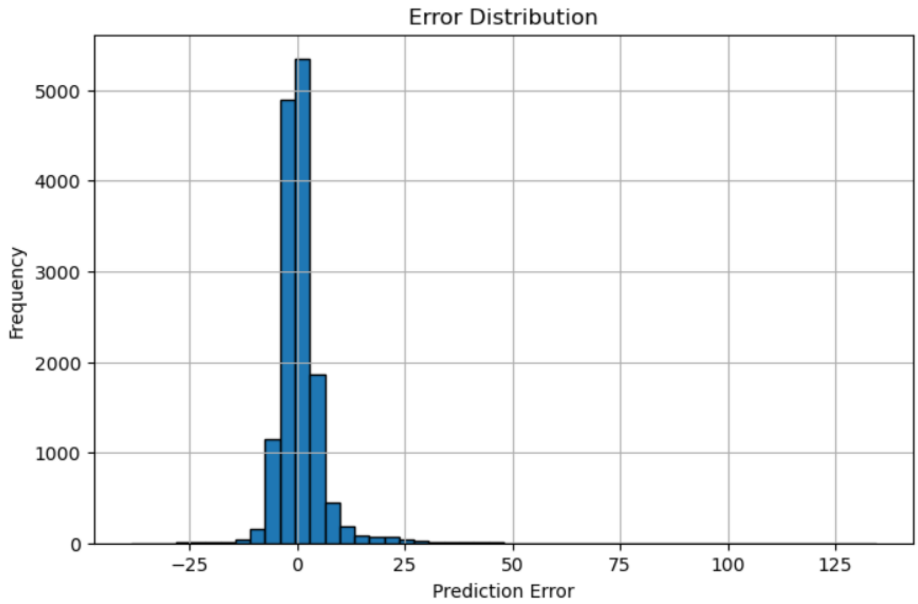
- Gradient Boosting Regressor (GBR) and LSTM models were developed using time-based splitting of the data — 70% for training and 30% for validation.
- Feature scaling was performed for the LSTM model using MinMaxScaler.
- LSTM input was reshaped to fit 3D sequential requirements.
- GridSearchCV was used to tune hyperparameters of GBR.

3. Model Evaluation:

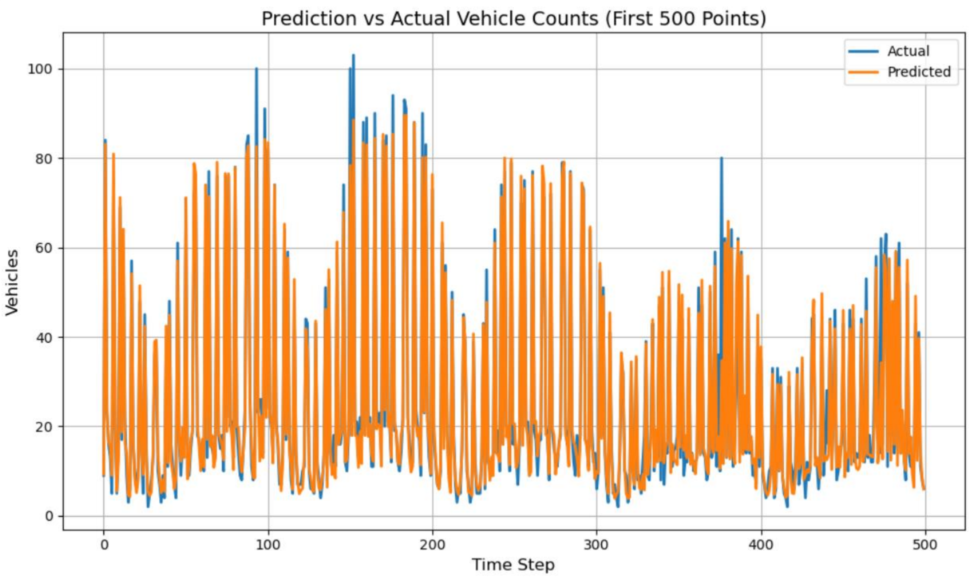
- Evaluation metrics used: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2 Score.
- GBR Validation Performance:
  RMSE ≈ *5.37*, MAE ≈ *3.14*, R^2 ≈ *0.959*
- LSTM Validation Performance:
  RMSE ≈ *5.13*, MAE ≈ 3.36, R^2 ≈ *0.962*
- Residual distribution was centered near zero, suggesting no major bias.



- Error distribution was approximately normal with no strong skew.



- The prediction vs. actual line plot showed good temporal alignment and relevance.



4. Cross-Validation Strategy:

- TimeSeriesSplit was applied with 5 folds for GBR.
- Cross-validated RMSE scores were printed and showed low variation.
- This confirmed stable generalization across time windows — essential for forecasting.
- Example: RMSE scores = [4.01900817, 5.3025713, 4.89756144, 4.91867063, 5.0225294], Average = 4.832068187263504

5. Model Refinement:

- Hyperparameter tuning (GBR):
    - n_estimators: [100, 150, 200]
    - max_depth: [3, 5]
    - learning_rate: [0.05, 0.1]
    - subsample: [1.0]
- Best Parameters:

  {'n_estimators': 100, 'max_depth': 3, 'learning_rate': 0.1, 'subsample': 1.0}

  These settings reduced validation error and avoided overfitting.

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
GridSearchCV
GridSearchCV(cv=3, estimator=GradientBoostingRegressor(random_state=42),
             n_jobs=-1,
             param_grid={'learning_rate': [0.05, 0.1], 'max_depth': [3, 5],
                         'n_estimators': [100, 150, 200], 'subsample': [1.0]},
             scoring='neg_mean_squared_error', verbose=1)
best_estimator_: GradientBoostingRegressor
GradientBoostingRegressor(random_state=42)
GradientBoostingRegressor
GradientBoostingRegressor(random_state=42)
```

6. Insights:

- Gradient Boosting and LSTM both performed well; LSTM handled temporal dependencies effectively.
- Time-based cross-validation proved essential for testing real-world generalization.
- Feature scaling and proper sequence shaping were critical for LSTM accuracy.