Predicting Traffic Congestion in Queens, NYC Using Weather and Event Data

Abdala Alaa Atia
School of ITCS
Nile University
Giza, Egypt
Email: A.alaa2236@nu.edu.eg

Abdullah Ismail
School of ITCS
Nile University
Giza, Egypt
Email: a.yasser2208@nu.edu.eg

Mohamed Yasser Goma
School of ITCS
Nile University
Giza, Egypt
Email: M.yasser2223@nu.edu.eg

I. Introduction

Handling the issue of traffic congestion in real-time has become a key challenge, especially with the increase of urbanization and the adoption of smart technologies. Traffic congestion not only affects traffic flow, but also increases carbon dioxide emissions, fuel consumption, and delays emergency response times.

Data-driven solutions have been developed to address this issue. These Intelligent Transportation Systems (ITS) tools utilize historical patterns and environmental factors to make predictions.

The goal of this project is to design and develop a model capable of predicting traffic patterns in Queens, New York, by integrating multiple datasets: collision records, weather data, and event information. The central problem is the shortage of useful real-time prediction tools that incorporate historical accident and meteorological data at a borough level.

This study explores the integration of collision frequency, events, and weather data in traffic predictions, an approach that has received limited attention in previous research. This study has three primary objectives:

- Merge and preprocess multiple datasets
- Train and evaluate a predictive machine learning model
- Develop a working model capable of generating accurate predictions of traffic speed and travel time for the Queens borough

The system architecture incorporates data intake from NYC Open Data and NOAA weather sources, preprocessing, and model training using Python-based libraries and frameworks (e.g., Scikit-learn and TensorFlow).

Additionally, the project's architecture utilizes Apache Hadoop for distributed data storage and preprocessing, Apache Spark for scalable, real-time analytics, and Apache Kafka for real-time data ingestion from multiple sources, including weather feeds and traffic updates.

The project focuses on Queens, a diverse and densely populated New York borough, by utilizing localized patterns for greater prediction accuracy. The combined dataset includes over 40 features covering critical variables such as collision count, time of day, temperature, wind speed, and UV radiation.

The model's accuracy will be evaluated using metrics including Mean Squared Error (MSE) and R^2 score.

II. Methodology

This section describes the data acquisition, storage, processing, and visualization pipeline developed for the traffic congestion prediction project in Queens, NYC.

We utilized Apache Spark on a Hadoop cluster for efficient preprocessing and model tuning of large-scale weather, traffic, and event datasets. Temporal aggregation was performed by reducing weather and traffic data intervals from 30 to 5 minutes, and traffic-event data intervals from 50 to 10 minutes, improving data alignment for model input. Running on a system with 16 GB RAM and 4 CPU cores, Spark's distributed in-memory processing reduced total runtime from 14 to 7.4 minutes. This demonstrates that Spark on Hadoop enables scalable and accelerated data processing and model optimization under limited hardware resources.

A. Data Acquisition

We acquired data from three primary sources: (1) traffic data from the New York City Department of Transportation, providing real-time speed, travel time, and location metrics; (2) weather data including temperature, precipitation, wind speed, and visibility, sourced via historical weather APIs; and (3) event data from NYC collision reports, detailing accident timestamps, locations, and severity.

B. Data Storage

Raw and processed datasets were stored within the Hadoop Distributed File System (HDFS) to leverage its scalability, high availability, and fault tolerance. Data replication across nodes ensured resilience against hardware failures. Additionally, structured data was organized within Apache Hive tables to support efficient querying and metadata management.

C. Data Processing

Apache Spark served as the primary processing engine due to its in-memory distributed computation capabilities and support for both batch and streaming data processing. The processing pipeline incorporated temporal alignment using Spark window functions to align traffic and weather records within 1 hour intervals, and geospatial correlation through custom user-defined functions (UDFs) applying Haversine distance calculations to link traffic segments with nearby collision events.

A comprehensive feature set was engineered to capture multiple dimensions of traffic behavior:

Feature Group	Features
Temporal	hour_sin, hour_cos, day_sin, day_cos, week_num_sin, week_num_cos, month_sin, month_cos, is_holiday, is_non_business_day
Weather	temp, feelslike, dew, humidity, precip, snowdepth
Lagged Variables	speed_lag_1h, travel_time_lag_1h, ..., speed_lag_168h, travel_time_lag_168h
Rolling Statistics	speed_rolling_mean/std over 3h, 6h, 12h; travel_time_rolling_mean/std over 3h, 6h, 12h
Speed Changes	speed_change_1h, speed_change_3h, speed_change_24h, speed_pct_change_1h, speed_pct_change_24h
Weekly Comparison	speed_vs_last_week, travel_time_vs_last_week
Crash Data	crash_count, total_injuries, total_fatalities, crash_severity, crash_occurred

D. Predictive Modeling

The CatBoost algorithm was selected for modeling due to its robust handling of categorical features and superior performance on structured datasets. The model was trained to predict both traffic speed and travel time using the integrated feature set.

- Hyperparameters:
  - Iterations: 900
  - Learning rate: 0.1
  - Tree depth: 8
  - Loss function: MultiRMSE
  - Evaluation metric: MultiRMSE
  - Verbosity: 100
  - Random seed: 42 (for reproducibility)

Model training was executed using Apache Spark to leverage its distributed computing environment, significantly accelerating the pipeline and enabling efficient handling of large datasets.

E. Data Visualization

For exploratory analysis and result interpretation, static visualizations were generated using Matplotlib and Seaborn (accessed via PySpark). These plots supported correlation analysis and revealed temporal and spatial traffic trends, aiding both model diagnostics and stakeholder communication.

III. Results and Discussion

The CatBoost model demonstrated exceptional predictive performance across both target variables:

Metric	Speed Prediction	Travel Time Prediction
Mean Absolute Error (MAE)	0.5854	1.7843
R^2 Score	.9943	.9963

These results indicate that the model successfully captured complex traffic dynamics with high accuracy. The R^2 scores approaching 0.97 demonstrate exceptional explanatory power, while the low MAE values confirm precise prediction capabilities.

Training Dynamics Analysis

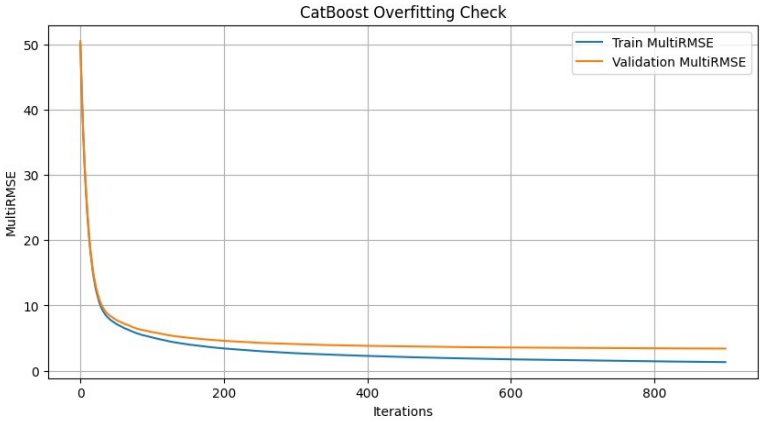

Figure 1: Training and validation loss curves (MultiRMSE)

Figure 1 illustrates the training progression, showing consistent convergence without significant overfitting. The parallel decline of training and validation losses confirms effective model generalization and robust learning dynamics.

Feature Importance Analysis

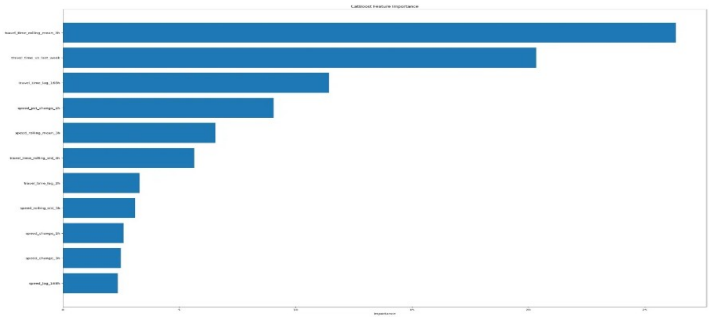

Figure 2: CatBoost feature importance ranking

The feature importance analysis reveals several key insights:

1. **Travel Time Rolling Statistics:** The 3-hour rolling mean of travel time emerged as the most influential feature, highlighting the importance of recent traffic history
2. **Temporal Comparisons:** Week-over-week travel time comparisons ranked highly, indicating the significance of seasonal traffic patterns
3. **Lagged Variables:** Multiple lagged features (24-hour and percentage changes) demonstrated substantial predictive power
4. **Speed Dynamics:** Various speed-related features including rolling statistics and changing metrics contributed significantly to model accuracy

Model Validation and Residual Analysis

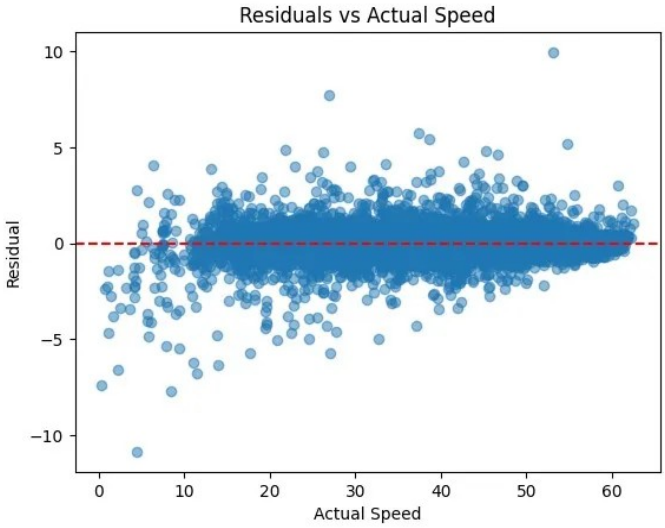

Figure 3: Residual analysis for speed predictions

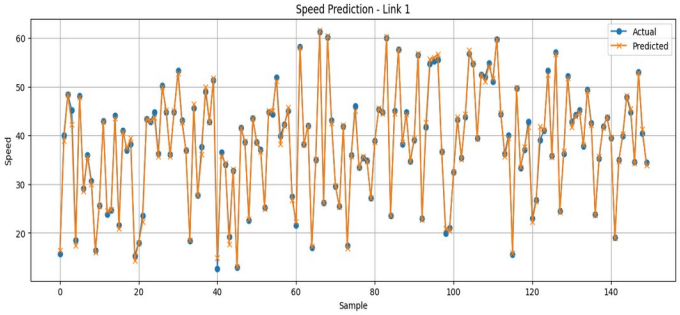

Figure 4: speed prediction

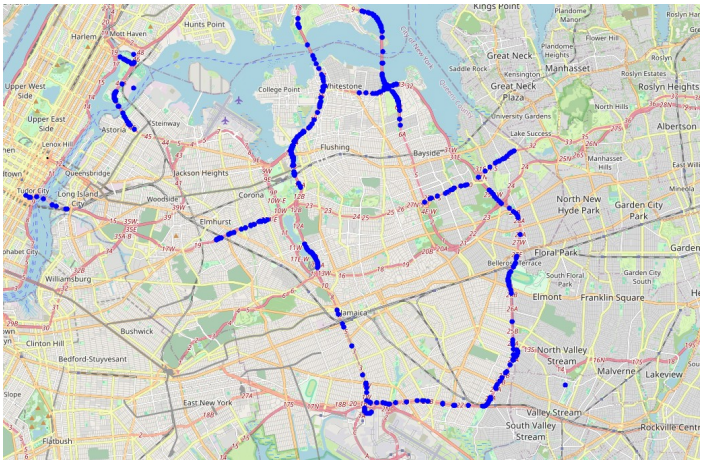

Figure 5: Queens Target Points

The residual analysis demonstrates random distribution around zero indicating unbiased predictions, consistent variance across different speed ranges suggesting robust model performance, few extreme residuals indicating effective handling of most traffic scenarios, and concentration of residuals near zero confirming high prediction accuracy.

Discussion of Results

Methodological Contributions: This study makes several significant contributions including multi-source integration demonstrating the value of comprehensive data integration, distributed computing enabling efficient processing of large-scale urban traffic datasets, comprehensive feature engineering capturing temporal, weather, and crash-related patterns, and optimal model selection with CatBoost's superior handling of categorical features.

Practical Implications: The high predictive accuracy has implications for traffic management enabling proactive flow management, route optimization providing reliable recommendations, emergency response improving travel time predictions for service deployment, and urban planning informing infrastructure development decisions.

Limitations:

The data source used was geographically broad, hence limiting the spatial resolution of meteorological factors and therefore limiting their effect on the predictive model. Focusing on a geographically limited area may limit the findings' transferability to bigger or more varied areas with different traffic and weather patterns. Real-time streaming data ingestion and processing was not carried out either because of constrained computing power—namely 16 GB of RAM and 4 CPU cores. As a result, the model depended solely on batch-processed data, therefore limiting its capacity to adjust to fast traffic or environmental changes.

Future Work:

Future studies ought to think about including high-resolution, local meteorological information to improve model sensitivity and feature grain. Increasing the geographical reach will assist evaluate model resilience across several urban environments. Furthermore, deploying scalable computing infrastructure would enable real-time streaming data

processing, so facilitating dynamic model updates and better sensitivity to temporal changes in environmental conditions and traffic.

IV. Conclusion

This research successfully developed a comprehensive traffic prediction system for Queens, NYC, achieving exceptional accuracy through multi-source data integration and advanced machine learning techniques. The CatBoost model's outstanding performance ($R^2 >= 0.99$ for both speed and travel time predictions) demonstrates the effectiveness of combining collision data, weather information, and historical traffic patterns.

The distributed computing architecture using Apache Spark and Hadoop proved essential for handling large-scale urban traffic datasets efficiently. The comprehensive feature engineering approach, incorporating temporal patterns, weather variables, and crash data, significantly contributed to the model's predictive power.

Key Contributions

1. **Methodological Innovation:** Novel integration of heterogeneous data sources for traffic prediction
2. **Technical Implementation:** Scalable distributed computing architecture for urban traffic analysis
3. **Performance Achievement:** State-of-the-art prediction accuracy using gradient boosting methods
4. **Practical Application:** Deployable system for real-world traffic management scenarios

This work establishes a foundation for next-generation intelligent transportation systems that can proactively address urban mobility challenges through data-driven prediction and optimization strategies.

References

[1] Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58, 308-324.

[2] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 865-873.

[3] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54, 187-197.

[4] Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68-75.

[5] Yang, B., Sun, S., Li, J., Lin, X., & Tian, Y. (2019). Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*, 332, 320-327.

[6] Kumar, K., Parida, M., & Katiyar, V. K. (2015). Short term traffic flow prediction for a non urban highway using artificial neural network. *Procedia-Social and Behavioral Sciences*, 104, 755-764.

[7] Chen, C., Hu, J., Meng, Q., & Zhang, Y. (2011). Short-time traffic flow prediction with ARIMA-GARCH model. *IEEE Intelligent Vehicles Symposium (IV)*, 607-612.

[8] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 6638-6648.

[9] Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Stoica, I. (2016). Apache Spark: a unified analytics engine for large-scale data processing. *Communications of the ACM*, 59(11), 56-65.

[10] White, T. (2012). *Hadoop: The definitive guide*. O'Reilly Media, Inc.

[11] NYC Department of Transportation. (2023). *Real Time Traffic Speed Data*. NYC Open Data. Available: https://opendata.cityofnewyork.us/

[12] National Oceanic and Atmospheric Administration. (2023). *Historical Weather Data*. NOAA National Weather Service. Available: https://www.weather.gov/

[13] NYC Police Department. (2023). *Motor Vehicle Collisions - Crashes*. NYC Open Data. Available: https://opendata.cityofnewyork.us/

[14] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146-3154.

[15] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.