Hourly Traffic Volume Prediction

Ameya Gurjar
School of Computing and
Augmented Intelligence
Arizona State University
Tempe Arizona USA
agurjar2@asu.edu

Anirudh Herady
School of Computing and
Augmented Intelligence
Arizona State University
Tempe Arizona USA
aherady@asu.edu

Arnav Rokde
School of Computing and
Augmented Intelligence
Arizona State University
Tempe Arizona USA
arokde@asu.edu

Bhaskar Bose
School of Computing and
Augmented Intelligence
Arizona State University
Tempe Arizona USA
bbose3@asu.edu

Avaneesh Shetti
School of Computing and
Augmented Intelligence
Arizona State University
Tempe Arizona USA
ashetti@asu.edu

ABSTRACT

Congestion management and urban planning are one of the most crucial problems to solve in the area of utilizing urban infrastructure, and we aim to focus on the forecasting of traffic volume on interstate highways specifically. However, prediction tasks are complicated as traffic data tends to show low correlation with standard weather and environment variables. Our approach tries to capture non-linear relationships in real-world traffic data that are helpful when input features don't show a strong correlation with the target. We implement models as LGBMRegressor, Random Forest, XGBoost, and Support Vector Regressor on our metro_interstate_traffic volume[4] dataset. For handling missing data, we utilize approaches such as Principal Component Analysis(PCA) for dimensionality reduction, Label Encoding for categorical variables, and iterative imputation. During our study, we observed that tree-based ensemble methods like XGBoost and LGBMRegressor resulted in better accuracy and generalization than the other models, this superior performance is due to efficient handling of non-linear patterns, higher robustness to noise or weakly correlated features, being better suited for high dimensional data and requiring lesser hyperparameter tuning as compared to deep learning models.

1 Introduction

Traffic volume forecasting is an essential component in modern intelligent transportation systems. With an increase in urbanization and vehicle density, the prediction of traffic patterns can significantly enhance road usage efficiency, reduce congestion, lower emissions and improve commuter experience. Multiple forecasting models can be used for aiding in traffic control systems, urban planning authorities and emergency response units in making data driven decisions.

The main issue here is predicting traffic volume at a fine temporal resolution such as hourly counts poses considerable challenges. Traffic data is noisy from the get go, often impacted by a multitude of dynamic factors such as weather conditions, time of day, public holidays, accidents as well as infrastructure changes. Furthermore, in our study we observed that environmental and temporal indicators displayed low correlation with the target variable and traffic volume. This low correlation complicated the application of traditional regression models and hence required careful preprocessing, feature engineering and model selection.

The methods already being used for traffic prediction range from the classical time series models like ARIMA to more recent machine learning based regressors such as Random Forests, Support Vector Regression and Gradient Boosted Trees. Also recently deep learning models have also shown promise, our focus in this study is on interpretable, general-purpose regressors that are computationally efficient and easier to deploy in practical systems.

In this project, we evaluated several regression algorithms starting from Random Forest Regressor, Support Vector Regressor, XGBoost, and LGBMRegressor on a real-world traffic dataset. The main content of the dataset is hourly traffic data recorded over several years, along with contextual features such as weather variables and time indices. We apply a range of preprocessing

steps, including one-hot encoding of categorical variables, dimensionality reduction using Principal Component Analysis, and iterative imputation to handle missing values.

The goal of this study is to benchmark model performance in a realistic, low signal setting and identify preprocessing techniques that can improve generalization. Through our experiments, we wanted to analyze how different regressors performed under specific conditions, how the preprocessing adds value, and what limitations persist despite these changes. Ultimately, what our findings suggest is that tree-based ensemble methods such as XGBoost and LGBMRegressor offer competitive performance, but the broader challenge of low feature correlation requires more advanced temporal modeling and feature enrichment.

2 Related Work

Traffic forecasting has experienced some substantial transformation, having shifted from classical time series approaches to advanced ML methodologies. Early research, such as work by Williams and Hoel [1], has shown that seasonal ARIMA models can effectively predict univariate traffic flow by modeling certain weekly trends. Despite their apparent strengths, these models many a time fall short in handling sudden traffic variations and other external influences such as weather.

The emergence of deep learning has enhanced predictive accuracy by capturing both spatial/temporal dependencies. For example, Zheng et al. [2] highlighted that models like LSTMs and CNNs can recognize complex traffic patterns. Although these methods often neglect environmental factors like weather. To address this problem, Hou et al. integrated weather data into a stacked autoencoder paired with a radial basis function neural network, achieving better accuracy at the cost of some increased computational demand.

More recent research focused on merging multiple data sources to improve traffic prediction. Studies by Wu et al. and Sivamurugan et al. have demonstrated the benefits of including weather and air quality data in deep learning models, emphasizing the role of environmental variables. Likewise, Abduljabbar et al. [3] found that while weather contributed only modestly to predictions on certain roads, it still enhanced the overall effectiveness of the model.

There has also been growing interest in multiple hybrid modeling techniques. Cheng et al. [2] proposed a LightGBM-GRU framework, while Zhang and Zhang [2] introduced an LSTM-XGBoost model, both of which combine feature selection with temporal modeling to enhance learning performance. However, some issues involving scalability persist. In response, our study aims to evaluate model generalization across various climatic conditions, and we test various methods on this data.

3 Dataset Description

3.1 Dataset Overview

The dataset used in this project is interstate_traffic.csv, comprising 48,204 rows and 9 columns. It captures hourly traffic volume data along with corresponding weather conditions over several years. Each record consists of temporal weather-related variables and traffic observations for a specific hour. The key columns in the dataset are:

1. **temp:** Temperature in Kelvin.
2. **rain_1h:** Amount of rain in the last hour (in mm).
3. **snow_1h:** Amount of snow in the last hour (in mm).
4. **clouds_all :** Percentage cloud cover.
5. **weather_main:** Categorical description of general weather (e.g., Clouds, Rain).
6. **weather_description**: More detailed weather descriptions (e.g., scattered clouds).
7. **date_time:** Timestamp of the recorded data (hour-level granularity).
8. **traffic_volume:** Number of cars on the road during the hour.

The column holiday had almost all values missing (48,143 out of 48,204), rendering it useless for analysis, and hence it was dropped from the dataset. Figure below summarizes the correlations between all numerical variables in the cleaned dataset.

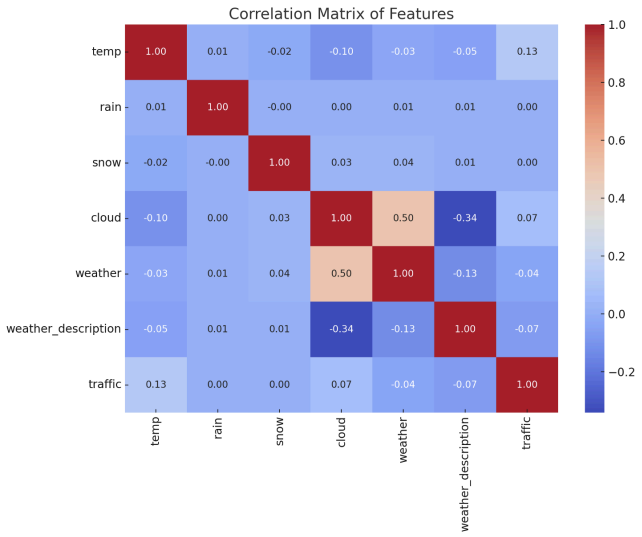

Figure 1: Correlation Matrix of Features

3.2 Preprocessing

Initial preprocessing steps were crucial to prepare the data for analysis and modeling:

1. **Missing Values:** As noted, the holiday column had excessive missing values and was dropped. Other

columns had complete data, so no further imputation was necessary.

2. **Renaming Columns:** Several columns were renamed for readability and consistency (rain_1h → rain, snow_1h → snow, clouds_all → cloud, weather_main → weather, and traffic_volume → traffic).
3. **Encoding Categorical Features:** The weather and weather_description columns were label-encoded to convert them into numerical values, which are necessary for downstream machine learning models.
4. **Datetime Parsing:** The date_time column was parsed into a proper datetime format and used later for time-series processing and modeling.

These preprocessing steps ensured that the dataset was clean, structured, and compatible with both statistical analysis and machine learning models.

3.3 Correlation and Dimensionality Reduction

An initial correlation matrix was computed to understand relationships between traffic and other features. Surprisingly, it revealed a very weak correlation between traffic volume and weather-related features such as temperature, rain, snow, cloud cover, and weather categories.

To explore latent interactions and reduce dimensionality, Principal Component Analysis (PCA) was applied to the scaled feature set. Two principal components (PCA1 and PCA2) were extracted and visualized against traffic.

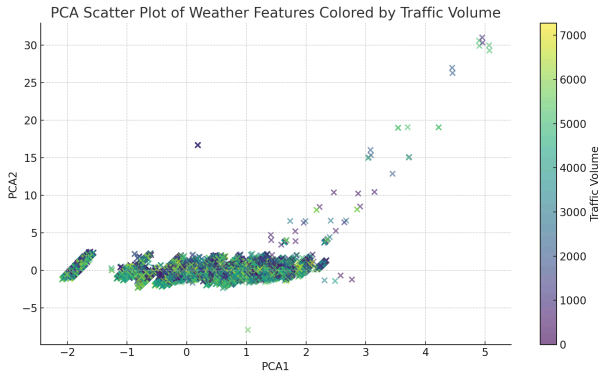

Figure 2: PCA Scatter Plot of Weather Features Colored by Traffic Volume

However, even these components demonstrated minimal correlation with traffic, as confirmed by the correlation heatmap of the PCA-transformed dataset.

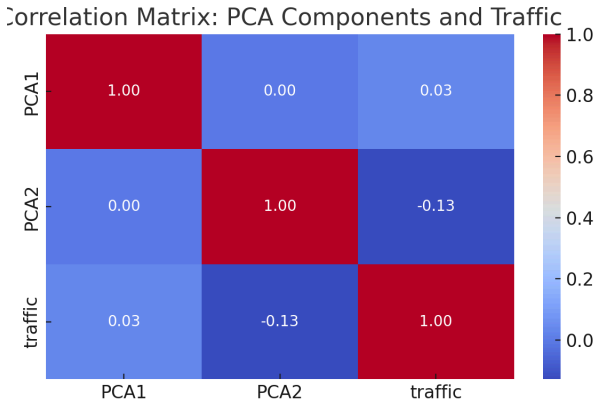

Figure 3:Correlation Matrix: PCA Components and Traffic

This observation motivated the need for a different modeling strategy—one that leverages time dependencies rather than external environmental features.

3.4 Motivation for Autoregressive Modeling

Given the low correlation between traffic volume and weather or PCA components, it was evident that traffic patterns were primarily time-dependent. This led us to adopt autoregressive models—models that use past traffic values (lags) to predict future values.

A detailed autocorrelation analysis of the traffic time series showed significant autocorrelation up to several lags. We defined a threshold of ±0.25 to determine the cut-off for statistically relevant lags, based on the autocorrelation function (ACF) plot. It was observed that using the past 8 hours (lags) captured meaningful temporal dependencies in the data without introducing excessive noise or overfitting.

Thus, we used the last 8 traffic observations as lag features in our autoregressive models (such as LSTM, GRU, Bi-LSTM, and CNN-LSTM). This approach allowed our models to focus purely on historical traffic trends, which proved to be a better predictor than weather-based features.

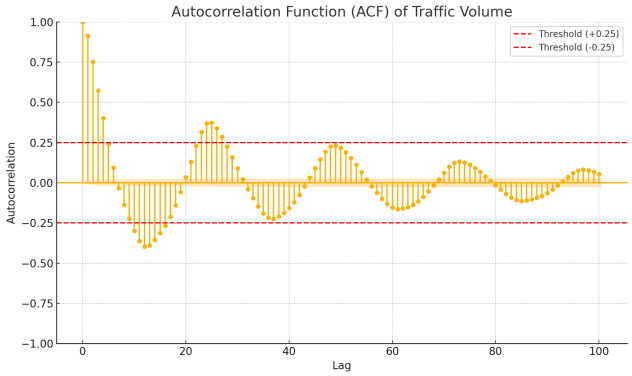

Figure 4: Autocorrelation Function (ACF) of Traffic Volume

4 Model Evaluation and Results

As part of our overall experimentation, we use the following models for our experiments -

- **GRU –** A variation of recurrent neural network that captures temporal dependencies with fewer parameters than LSTM.
- **LSTM –** Another type of RNN designed to learn long-term dependencies using gated cells to control information flow.
- **CNN-LSTM –** A hybrid model that extracts spatial features with CNN layers and learns other temporal patterns using LSTM layers.
- **Bi-Directional LSTM –** An LSTM that processes input sequences in forward and backward directions to capture context from both past and future.
- **XGBoost –** An efficient and scalable implementation of gradient boosting that builds an ensemble of decision trees.
- **Random Forest –** An ensemble learning method that builds multiple decision trees and combines their outputs for overall classification.
- **SVR –** A model performing regression involving Support Vector Machines that fits the best possible line among data points.
- **Gradient Boosting –** A ML technique that sequentially builds models to correct the errors of prior models, improving performance over time.
- **LGBM –** A fast and efficient gradient boosting framework based on decision tree algorithms, it is optimized for speed and memory usage.

4.1 Training Procedure

To assess the predictive performance of each model, we trained them individually on the dataset, focusing on their effectiveness in forecasting traffic volume. Following a Batch normalization, all models were trained using a batch size of 32, the Adam optimizer for gradient updates, and a fixed learning rate of 0.001.

To prevent overfitting and ensure convergence of the training loss, we employed early stopping based on validation performance. During training, model weights were saved only when an improvement in validation metrics was observed, ensuring the retention of the best-performing parameters.

4.2 Results

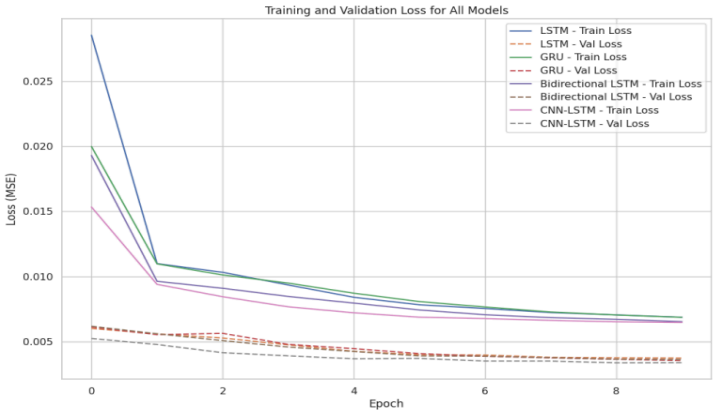

Fig 1 - Training/Validation Loss per epoch for RNN-Based methods

Model	MSE	MAE	RMSE	MAPE (%)
LSTM	193,744.95	310.62	440.16	15.19
GRU	181,909.84	295.87	426.51	14.60
Bidirectional LSTM	188,251.44	304.95	433.88	14.42
CNN-LSTM	173,984.27	286.50	417.11	13.63

Fig 2 - Validation metrics on the test set for RNN-Based methods

The corresponding graphs and tables evaluate the performance of the RNN-based methods. Fig 1, shows the loss graphs (MSE Loss) where the hybrid model CNN-LSTM gives the lowest loss values for both the training and validation sets. The table in Fig 2, shows validation metrics used to compare the RNN-Based methods on the test set. For all the metrics, the hybrid model (CNN-LSTM) gives the best results.

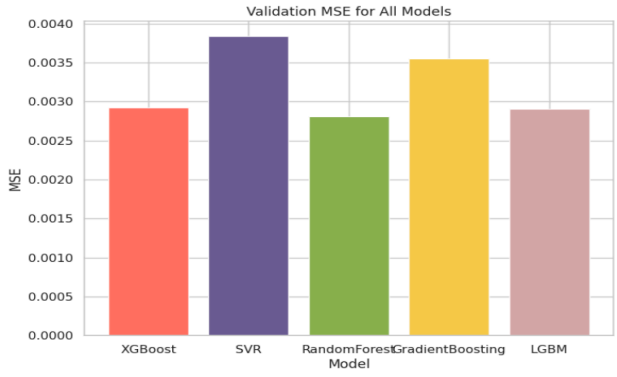

Fig 3 - Validation MSE for ML-Based methods (Barchart)

Model	MSE	MAE	RMSE	MAI
XGBoost	161,079.29	258.62	401.35	12.4
SVR	200,588.88	340.43	447.87	24.0
Random Forest	154,804.30	251.14	393.45	11.8
Gradient Boosting	184,745.64	296.26	429.82	15.2
LGBM	152,882.19	258.94	391.00	12.8

Fig 4 - Additional validation metrics for ML-Based methods

The barchart and the table in Fig 3 and Fig 4, show us the relative performance of the Machine Learning methods with respect to each other. As per their performance, LGBM, a boosting method provides the best performance on the test set. If we consider the overall performance for both RNN-Based and ML methods, we see that the LGBM performs the best.

4.3 Significance of Results

The dataset provided an understanding of how traffic volume is related to various features in a tabular form. It contained no sequential dependencies, and contained non-linear relationships among the co-variate values. Through our experimentation, we discovered that Boosting and Bagging techniques gave the best performance. And of these methods, LGBM performed the best on the test set.

Looking carefully, since ML Based methods do not assume a sequential dependence on the data, they performed much better then their RNN counterparts. RNNs are designed for sequential dependence and hence are excellent choices when temporal dependencies are involved. However, given the data, it was not suitable for such an assumption.

Among the boosting/bagging methods, due to LGBM's Leaf-wise growth which tends to reduce loss more aggressively then level wise growth (in other boosting techniques) it resulted in deeper, more expressive trees. And this helped capture more complex dependencies in better ways.

REFERENCES

[1] B. M. Williams and L. A. Hoel, "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[2] Z. Zheng, L. Yang, and H. Cheng, "Deep Learning for Traffic Flow Prediction: A Comprehensive Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4654–4671, 2019.

[3] R. Abduljabbar, H. Dia, and S. Liyanage, "Machine Learning Models for Traffic Prediction on Arterial Roads Using Traffic Features and Weather Information," *Applied Sciences*, vol. 14, no. 23, p. 11047, 2024.

[4] J. Hogue. "Metro Interstate Traffic Volume," UCI Machine Learning Repository, 2019. [Online]. Available: https://doi.org/10.24432/C5X60B.

[5] Mosavi, S. (n.d.). LSTW: A Large-Scale Dataset for Long-Short-Term Web Page Revisit Prediction. Retrieved from https://smoosavi.org/datasets/lstw.

[6] City of New York. 2024. Traffic Volume Counts. NYC Open Data. Available at: https://data.cityofnewyork.us/Transportation/TrafficVolume-Counts/btm5-ppia.

[7] City of Phoenix, "Street Traffic Volumes," *Phoenix Open Data Portal*, 2025. [Online]. Available: https://www.phoenixopendata.com/dataset/street-traffic-volumes

[8] City of Chicago, "Chicago Traffic Tracker - Congestion Estimates by Segments," *Chicago Data Portal*, 2025. [Online]. Available: https://data.cityofchicago.org/Transportation/ChicagoTraffic-Tracker-Congestion-Estimates-bySe/n4j6-wkk f