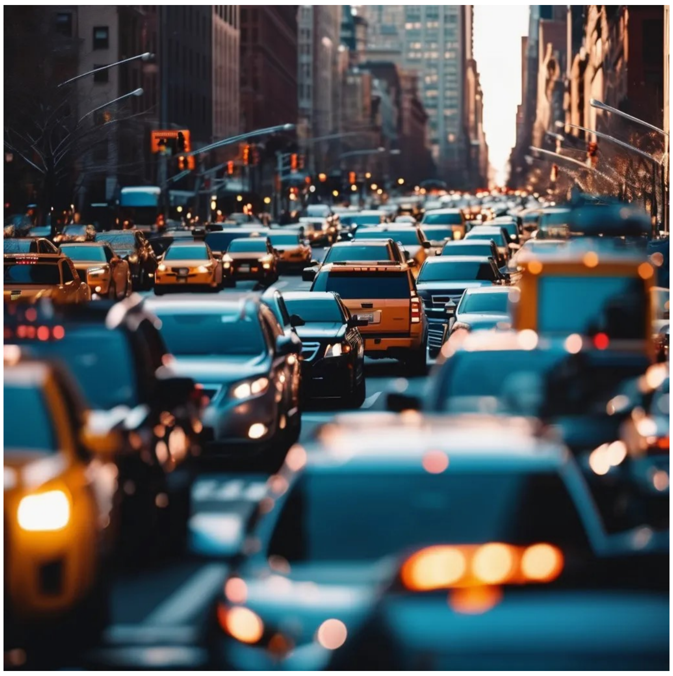
Intelligent Systems
Deep Learning study
Lisboa, Portugal, Feb 03, 2025
João Sousa, Rodrigo Ventura (Professors)

Traffic Volume Prediction Using Deep Learning

Francisco Rosado de Carvalho
Affiliation of the Author
IST 111000
francisco.r.carvalho@tecnico.ulisboa.pt



This image was generated using PIXLR

Contents

1. Project Overview

Traffic volume prediction is a critical component of modern urban planning, transportation management, and smart city initiatives. Accurate forecasting models can help reduce congestion, optimize traffic flow, and enhance commuter experience by providing real-time insights into traffic conditions. With the increasing availability of traffic-related data, machine learning has emerged as a powerful tool for predicting traffic patterns based on historical data and external influencing factors.

This project explores various machine learning approaches for traffic volume prediction, utilizing a dataset containing multiple environmental and temporal features. The primary objective is to compare the effectiveness of different models, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Temporal Convolutional Network (TCN), and Extreme Gradient Boosting (XGBoost). These models were chosen due to their strong performance in time-series forecasting and structured data analysis.

The complete code for this project is available in a Jupyter Notebook, which includes data preprocessing, model training, and evaluation processes. You can access the notebook and check the dataset through the following links:

GitHub Repository: `https://github.com/FranciscoCarvalho26/Traffic-Volume-Project`

Dataset DOI: `https://www.kaggle.com/datasets/rohith203/traffic-volume-dataset/data`

2. Exploratory Data Analysis

2.1 Data Overview

The dataset consists of 33,750 entries with 15 columns, representing different weather and traffic conditions collected over time. Each entry corresponds to hourly observations of various environmental parameters and traffic volume. Below is a brief summary of the data:

- **Datatime:** The dataset is recorded with a timestamp, starting from October 2, 2012, and ending on May 17, 2017. This time frame provides a comprehensive look at traffic and weather conditions over several years.
- **Weather Data:**
  - **Air Pollution Index:** Ranges from 10 to 299, with a mean value of 154.8, representing air quality at the time of observation.
  - **Humidity:** Ranges from 13% to 100%, with an average of 71.2%. It captures the moisture content in the air.
  - **Wind Speed:** Varies between 0 and 16 mph, with and average value of 3.38 mph.
  - **Wind Direction:** The direction of the wind is measured in degrees (0 to 360), with an average value of 199.47 degrees.
  - **Visibility in Miles:** Ranges from 1 to 9 miles, with an average visibility of about 5 miles.
  - **Dew Point:** The dew point ranges from 1°C to 9°C, indicating the temperature at which water vapor in the air condenses.
  - **Temperature:** Ranges from 271.72 K to 308.24 K, with an average temperature of 280.07 K.
  - **Rainfall and Snowfall:** Both rain and snow are recorded as precipitation rates in inches per hour. On average, rainfall is observed at a rate of 0.45 inches per hour, and snowfall is much less frequent with a rate of 0.0003 inches per hour.
  - **Cloud Cover:** The cloud cover ranges from 0% to 100%, with a mean of 50.46%.
  - **Weather Description:** This categorical variable contains descriptions of the weather, such as "Clouds", "Clear" and "Rain".
- **Traffic Volume:** The traffic volume ranges from 0 to 7,280 vehicles per hour, with an average of approximately 3,240 vehicles per hour.

There are some additional observations to note, for example the column **"is_holiday"** contains 43 non-null values and indicates whether a particular observation falls on a holiday. Rows where this value is null simply represent non-holiday entries, ensuring that non-holiday traffic patterns are the primary focus of the dataset. Weather-related variables have different ranges, reflecting the different conditions during the study period. Finally, traffic volume, the target variable, shows considerable variation, highlighting periods of both low and high traffic.

2.2 Individual Feature Analysis

The distributions of the numerical and categorical characteristics were examined using histograms and bar plots to provide insight into their characteristics:

- **Numerical Features**

  The histogram illustrated below shows the distributions of features such as air pollution index, humidity, wind, speed, visibility, and traffic volume. This visualizations highlight key patterns like skewness and clustering.

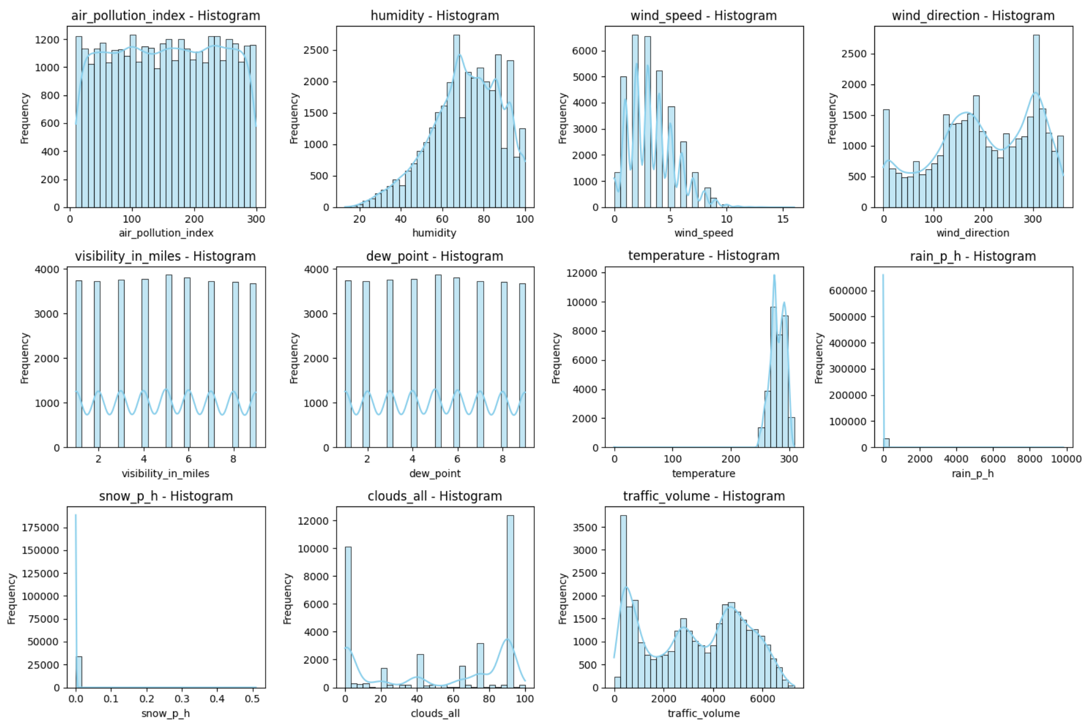


Figure 1: Histograms for All Numerical Features.

  The numerical features exhibit diverse distributions. Some features, such as air pollution index and wind direction, show uniformity, while others, like humidity and temperature, are positively skewed.

  Cloud cover shows a bimodal distribution, with peaks at 0% and 100%, indicating the dominance of clear or overcast conditions.

  Traffic Volume demonstrates distinct peaks, corresponding to high and low traffic periods.

  Rainfall and Snowfall are both part of the numerical analysis, given their nature as continuous variables related to weather conditions. Their occurrence is relatively rare, and understanding these variables helps assess their potential impact on traffic volume. For that purpose, the following bar plot shows the percentage of hours with recorded rainfall and snowfall, emphasizing their rarity.

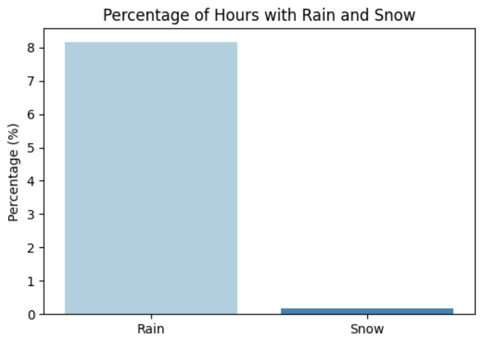


Figure 2: Rainfall and Snowfall Occurrence.

Due to their infrequency and low values, these variables are unlikely to significantly influence traffic volume predictions.

- **Categorical Features**

  The following bar plots include distributions of weather descriptions, holiday occurrences, and weather types to showcase their frequency and significance:

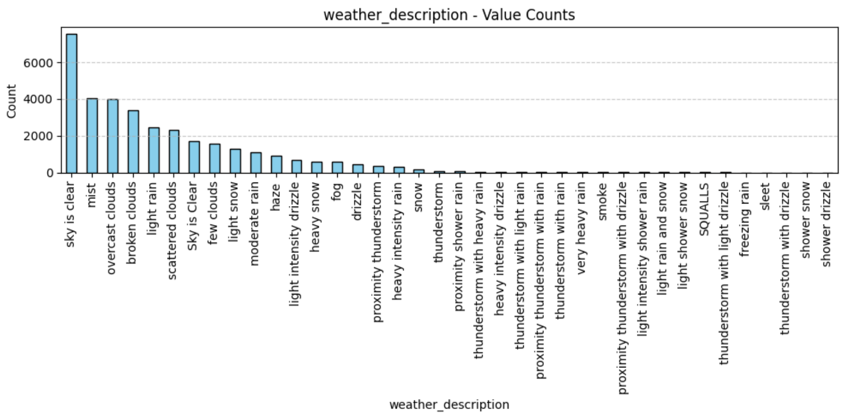


Figure 3: Bar Plot of Weather Descriptions.

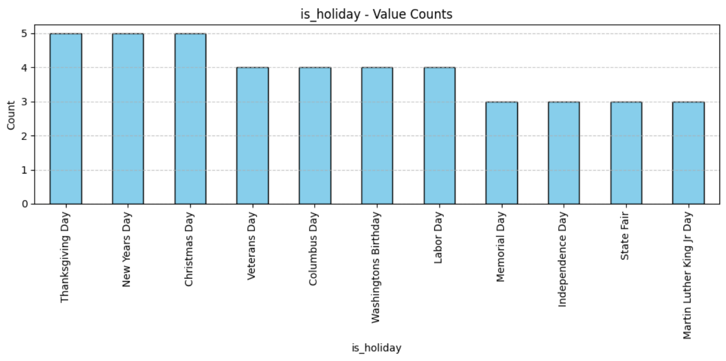


Figure 4: Bar Plot of Holidays.

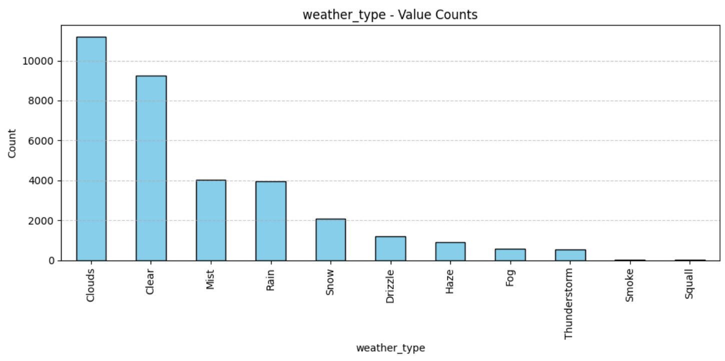
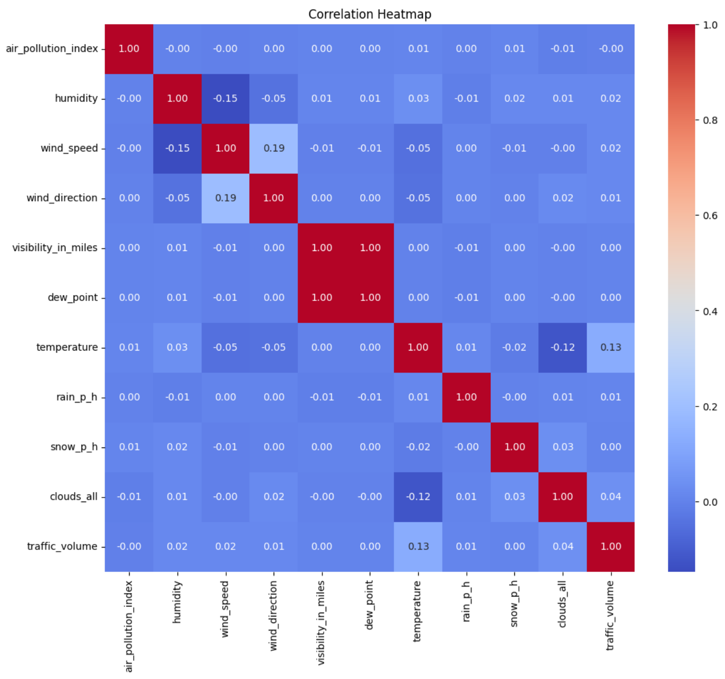


Figure 5: Bar Plot of Weather Types.

2.3 Correlation Analysis

A correlation was conducted to assess the relationships between the numerical features in the dataset. The correlation heatmap below shows the results of this analysis.



	air_pollution_index	humidity	wind_speed	wind_direction	visibility_in_miles	dew_point	temperature	rain_p_h	snow_p_h	clouds_all	traffic_volume
air_pollution_index	1.00	-0.00	-0.00	0.00	0.00	0.00	0.01	0.00	0.01	-0.01	-0.00
humidity	-0.00	1.00	-0.15	-0.05	0.01	0.01	0.03	-0.01	0.02	0.01	0.02
wind_speed	-0.00	-0.15	1.00	0.19	-0.01	-0.01	-0.05	0.00	-0.01	-0.00	0.02
wind_direction	0.00	-0.05	0.19	1.00	0.00	0.00	-0.05	0.00	0.00	0.02	0.01
visibility_in_miles	0.00	0.01	-0.01	0.00	1.00	1.00	0.00	-0.01	0.00	-0.00	0.00
dew_point	0.00	0.01	-0.01	0.00	1.00	1.00	0.00	-0.01	0.00	-0.00	0.00
temperature	0.01	0.03	-0.05	-0.05	0.00	0.00	1.00	0.01	-0.02	-0.12	0.13
rain_p_h	0.00	-0.01	0.00	0.00	-0.01	-0.01	0.01	1.00	-0.00	0.01	0.01
snow_p_h	0.01	0.02	-0.01	0.00	0.00	0.00	-0.02	-0.00	1.00	0.03	0.00
clouds_all	-0.01	0.01	-0.00	0.02	-0.00	-0.00	-0.12	0.01	0.03	1.00	0.04
traffic_volume	-0.00	0.02	0.02	0.01	0.00	0.00	0.13	0.01	0.00	0.04	1.00

Figure 6: Correlation heatmap showing the relationships between the numerical features.

From figure 6, it can be observed that some features have strong correlations, such as **visibility_in_miles** and **dew_point**, which have a perfect correlation of 1. As a result, **dew_point** was removed from further analysis to avoid multicollinearity issues. When examining the correlation with the target variable, **traffic_volume**, no strong correlations were found. This lack of strong correlation could be due to the complex, nonlinear factors influencing traffic volume that are not captured by these numerical features.

2.4 Categorical Feature Analysis: Kruskal-Wallis Test

Initially, an **ANOVA** test was performed to analyze the relationships between categorical and numerical features. ANOVA assumes normally distributed data, homogeneity of variances, and independent observations, but these assumptions were not met, and small sample sizes rendered the results unreliable.

To address this, the **Kruskal-Wallis** test, a non-parametric alternative suitable for small samples and non-normal data, was applied. In this test, the H-statistic measures group differences based on ranked data, and the p-value indicates the likelihood of observing the results under the null hypothesis (no group differences). Significant results were identified in the following cases:

- **Weather description:** Strong associations with numerical features like temperature ($H = 6194.83$. $p < 0.0001$), rain per hour ($H = 10088.22$, $p < 0.0001$), and traffic volume ($H = 666.82$, $p < 0.0001$).
- **Weather type:** Significant relationships with temperature ($H = 4049.87$, $p < 0.0001$) rain per hour ($H = 9402.05$, $p < 0.0001$), and traffic volume ($H = 433.41$, $p < 0.0001$).

2.5 Scatter Plot Analysis of Important Features

Three features were selected as important for understanding traffic volume: **clouds_all**, **wind_speed**, and **temperature**. Scatter plots were generated to illustrate their relationships with the target variable, **traffic_volume**. Below are the scatter plots for these features:

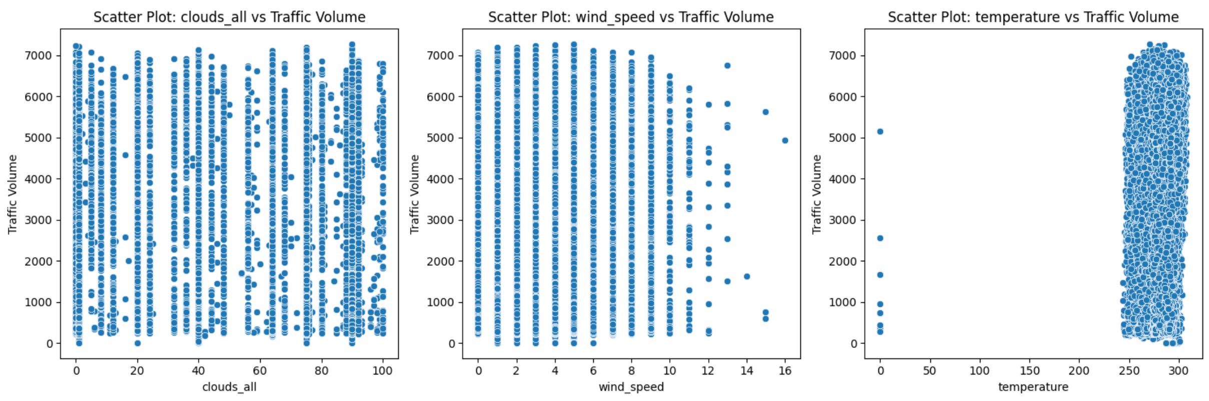


Figure 7: Scatter plots of traffic_volume vs. clouds_all, wind_speed, and temperature.

2.6 Time Series Analysis

Next, it was analyzed how traffic volume varied over time to identify any temporal patterns. This is important for understanding how traffic volume changes on different days, months, or seasons, which can be critical for traffic management and planing.

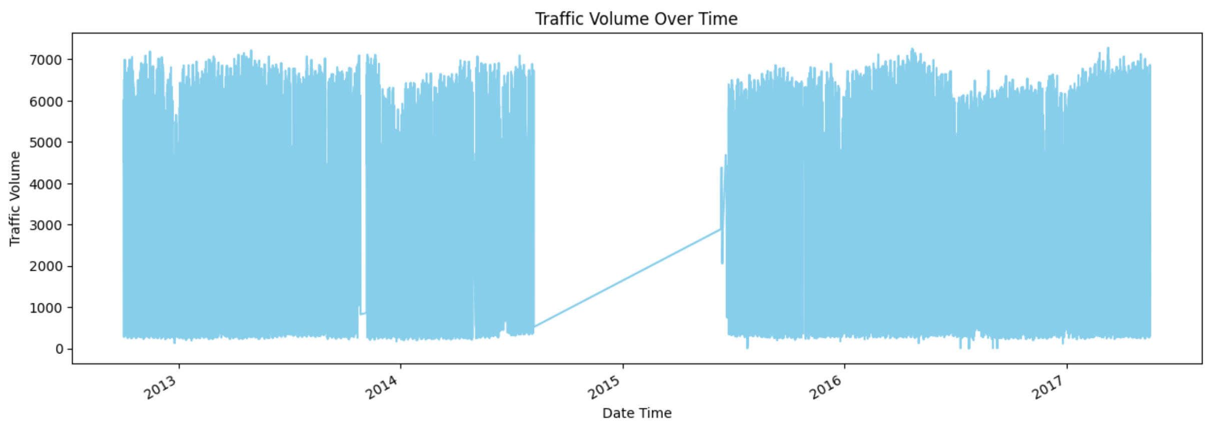


Figure 8: Times series plot showing traffic volume over time.

The time series plot demonstrates that traffic volume fluctuates significantly, indicating periods of higher and lower traffic. To further analyze this behavior, another time series analysis was conducted, but now it was limited to data from a one-year period to make the trends more visible and easier to analyze.

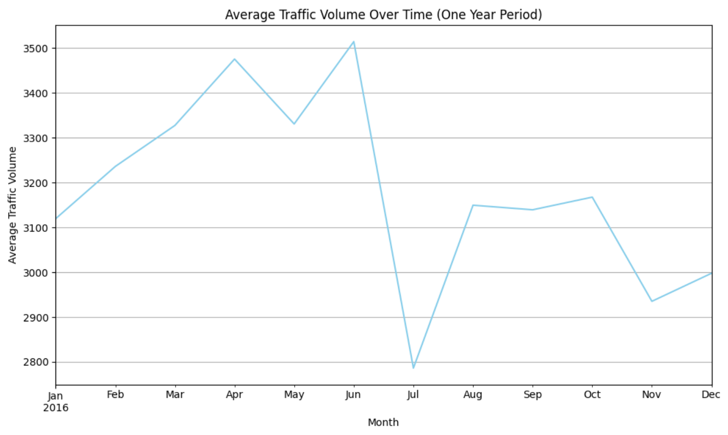


Figure 9: Times series plot showing traffic volume over one year period.

This plots shows the seasonal trends in traffic volume, with observable peaks during the first semester of the year.

2.7 Traffic Volume on Holidays

In this final analysis of the exploratory data phase, the variation in traffic volume across different types of holidays was examined. A box plot was used to visualize the differences in traffic patterns for varying holiday types. The analysis revealed that traffic volume differs significantly depending on the type of holiday, highlighting the influence of the holiday's nature on traffic behavior.

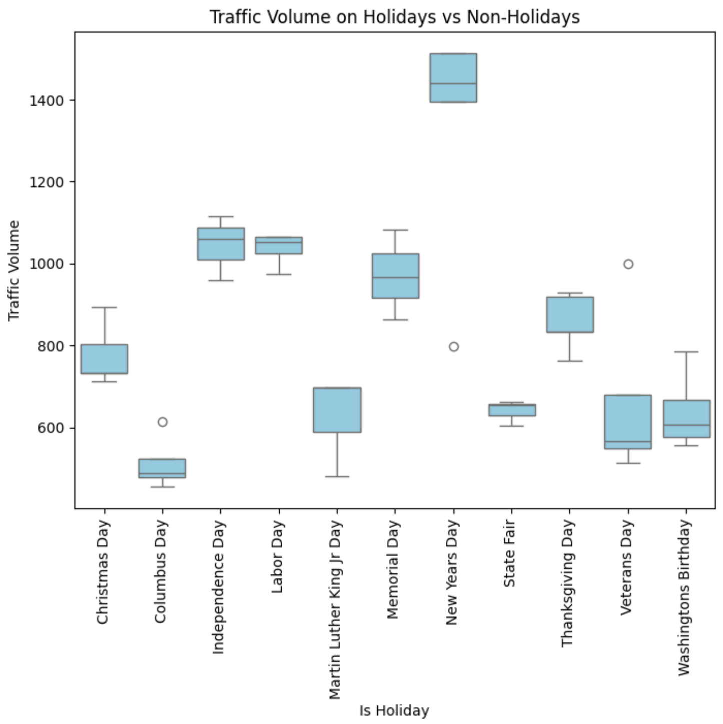


Figure 10: Box plot comparing traffic volume across holiday types.

3. Preprocessing

Data preprocessing is a crucial step in ensuring that the dataset is clean, structured, and suitable for training machine learning models. Proper preprocessing enhances model performance by reducing noise and improving generalization. Figure 11 illustrates the preprocessing pipeline, outlining the key transformations applied to the dataset before model training.

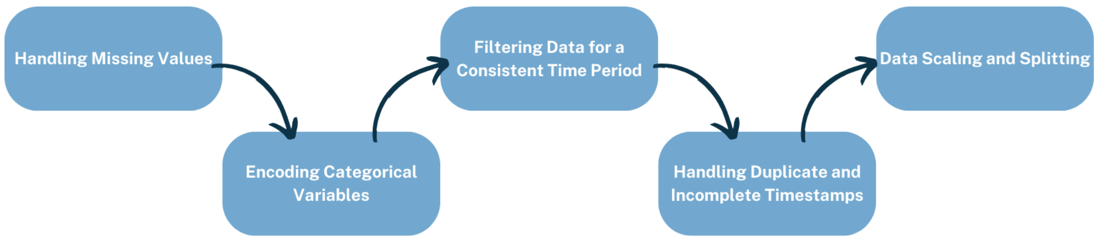

Handling Missing Values
Encoding Categorical Variables
Filtering Data for a Consistent Time Period
Handling Duplicate and Incomplete Timestamps
Data Scaling and Splitting

Figure 11: Preprocessing Pipeline.

3.1 Handling Missing Values and Encoding Categorical Variables

Handling Missing Values

An initial assessment of the dataset was conducted to determine whether any missing values were present. The analysis confirmed that the dataset did not contain any missing entries. However, despite the absence of explicit missing values, additional steps were necessary to ensure data consistency and quality before proceeding with model development.

Encoding Categorical Variables

Machine learning models require numerical inputs, therefore, categorical features in the dataset needed to be converted into numerical representations. The categorical variables **is_holiday**, **weather_type**, and **weather_description** were transformed using one-hot encoding. This approach creates binary variables for each unique category, ensuring that the categorical information is effectively represented without introducing an ordinal relationship where none exists.

Furthermore, to streamline the subsequent data processing steps, the target variable **traffic_volume** was repositioned as the last column in the dataset. This reordering facilitates efficient data partitioning when preparing training and testing subsets.

3.2 Temporal Filtering and Data Consistency

Filtering Data for a Consistent Time Period

Upon further inspection of the dataset, it was observed that a significant portion of data was missing from the early records, with nearly an entire year of observations absent. To maintain the temporal integrity of the dataset and avoid gaps that could affect model performance, data records prior to **January 1, 2016, at 00:00:00** were excluded. This ensured that only continuous and reliable data was used for training and evaluation.

Handling Duplicate and Incomplete Timestamps

A key challenge identified in the dataset was the presence of duplicate records for certain hourly timestamps. In some cases, multiple records existed for the same hour of a given day, each containing different feature values. This inconsistency could introduce noise into the model, leading to inaccurate predictions. To address this issue, the dataset was grouped by data and hour, and the mean values of numerical features were computed for each hour. This approach helped to retain a representative value for each hour while mitigating the impact of duplicate entries.

Additionally, it was observed that some hours were missing from the dataset. For instance, there were cases where the recorded timestamps skipped certain hours (e.g., a transition from 9:00 AM to 11:00 AM, omitting 10:00 AM). To correct this, structured approach was applied:

- The dataset was re-indexed to enforce an **hour frequency**, ensuring that all hours within the recorded period were represented.
- Any newly introduced missing values resulting from this re-indexing were filled using the **forward-fill method**, which propagates the most recent available value. This method was chosen as it preserves temporal trends while preventing unrealistic abrupt changes in the data.

Data Consistency and Rounding Adjustments

To further refine the dataset, numerical variables were rounded to appropriate decimal places, ensuring uniformity across records. For example:

- Integer values, such as **air pollution index**, **humidity**, **wind speed**, and **cloud coverage**, were rounded to whole numbers.
- Continuous variables, such as **temperature**, were retained with two decimal places to preserve precision while avoiding excessive granularity.

3.3 Data Scaling and Splitting

Standardization of Data

To ensure that features were appropriately scaled for model training, **StandardScaler** was applied to the dataset. Standardization transforms the features so that they have a mean of zero a standard deviation of one preventing features with large numerical ranges from dominating model training. This is particularly important for deep learning models, which are sensitive to varying features scales.

Spitting the Data for Model Training

To evaluate model performance effectively, the dataset was divided into training and testing subsets. A **90%-10% split** was used, with the majority of the data allocated for training while retaining a smaller portion for evaluation.

For **deep learning models** (LSTM, GRU, and TCN), a **time-series approach** was required, necessitating the transformation of the dataset into sequences. A **time step of 48 hours** was chosen, meaning that for each prediction, the models were with the preceding 48 hours of data. This transformation structured the dataset into three-dimensional input sequences suitable for recurrent and convolutional neural networks.

For **XGBoost**, a more traditional approach was used, where the dataset was split into feature matrices **(X_train, X_test)** and target vectors **(Y_train, Y_test)** without requiring sequence information. Cross-validation was utilized during hyperparameter tuning and feature selection to ensure robust performance evaluation.

In cases where a validation set was required, **10% of the training data** was set aside for validation purposes. However, since cross-validation was often incorporated into hyperparameter tuning and feature selection, the explicit creation of a validation set was not always necessary.

These preprocessing steps ensured that the dataset was properly structured, scaled, and partitioned, laying the groundwork for effective development and evaluation.

4. Model Development

The model development phase focuses on implementing and fine-tuning different machine learning architectures to predict traffic volume. Various models, including LSTM, GRU, TCN, and XGBoost, are trained and evaluated to compare their predictive capabilities. Figure 12 provides an overview of the modeling process, from data input to prediction generation.

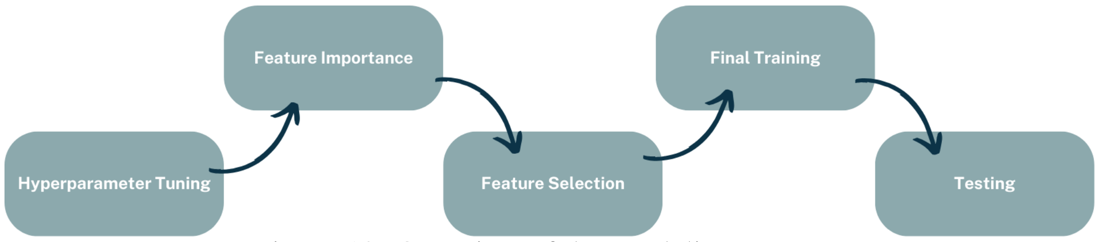


Figure 12: Overview of the Modeling process.

4.1 Long Short-Term Memory (LSTM)

4.1.1 Introduction to LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for modeling sequential data, particularly when long-term dependencies need to be captured. Unlike traditional RNNs, LSTMs effectively mitigate the vanishing gradient problem, allowing them to retain information over longer time periods. This characteristic makes LSTM an excellent choice for traffic volume prediction, where patterns and dependencies span across different time scales. Given the temporal nature of traffic data and the goal of accurate forecasting, LSTMs were selected as the first model for this study.

4.1.2 Hyperparameter Tuning

The first step in developing the LSTM model involved hyperparameter tuning to identify the optimal configuration for the network. A systematic grid search was conducted using the following hyperparameter options:

- **Number of LSTM Units:** [32, 64]
- **Learning Rates:** [0.00001, 0.00005, 0.0001]
- **Dropout Rates:** [0.2, 0.5]
- **Optimizers:** Adam, RMSprop
- **Activation Functions:** 'relu', 'tanh'
- **Number of Layers:** [1, 2, 3]

To evaluate the performance of different configurations, a 3-fold cross-validation approach was employed. The data was split into training and validation subsets, and each combination of hyperparameters was tested. The key steps of the hyperparameter tuning process included:

1. **Model Definition:** A function was implemented to dynamically create LSTM models based on the specified hyperparameters. The architecture allowed for up to three LSTM layers, each followed by a dropout layer to prevent overfitting.
2. **Cross-Validation:** For each combination of hyperparameters, the model was trained and validated across the folds. The validation loss for each fold was recorded to compute the average validation loss for the configuration.
3. **Model Selection:** The combination of hyperparameters yielding the lowest average validation loss was selected as the best configuration.

The best-performing model configuration was identified and saved with the following hyperparameters:

- **Number of Layers:** 3

- **LSTM Units per layer:** [32, 32, 32]
- **Optimizer:** Adam
- **Learning Rate:** 0.0001
- **Dropout Rate:** 0.2
- **Activation Function:** 'tanh'

4.1.3 Model Evaluation with the Best Configuration

After determining the best model configuration, its performance was evaluated using the test dataset. The saved model, with the best configuration, was loaded and used to make predictions for the last 24 hours in the test dataset. The predictions were compared with the actual traffic volume values, and the results were visualized and quantified using standard forecasting evaluation metrics.

1. **Performance Visualization:** The predicted and true traffic volume values were plotted to assess how well the model captured the temporal patterns. The comparison plot highlighted the alignment (or discrepancies) between the predictions and actual values over the selected timeframe.

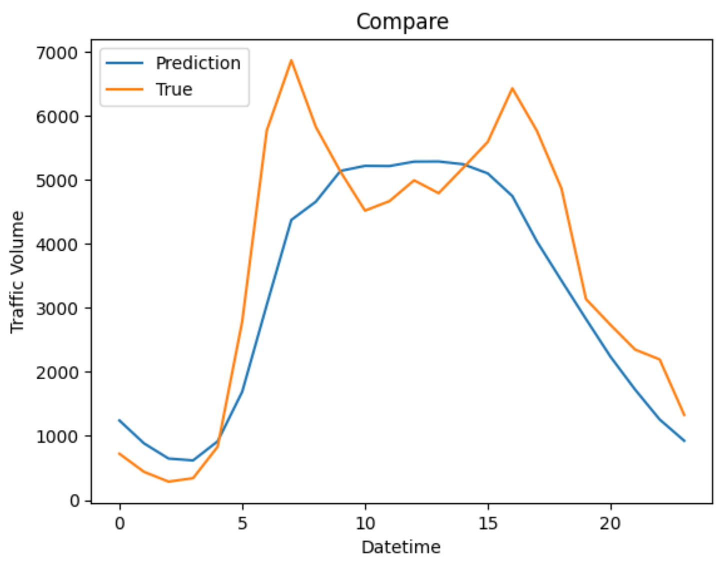


Figure 13: LSTM - First - Prediction vs. True Traffic Volume.

2. **Evaluation Metrics:** Several metrics were calculated to quantitatively assess the model's performance:

    - **Root Mean Squared Error (RMSE):** Evaluates the model's prediction error in the same units as the target variable.
    - **Mean Absolute Error (MAE):** Measures the average magnitude of errors in the predictions.
    - **Mean Absolute Percentage Error (MAPE):** Provides a percentage-based error measure.
    - **R-Squared (R²):** Indicates the proportion of variance in the target variable explained by the model.

    The evaluation results were summarized in two bar plots:

- Scale-dependent errors (e.g., RMSE, MAE)
- Percentage errors (e.g., MAPE, R²)

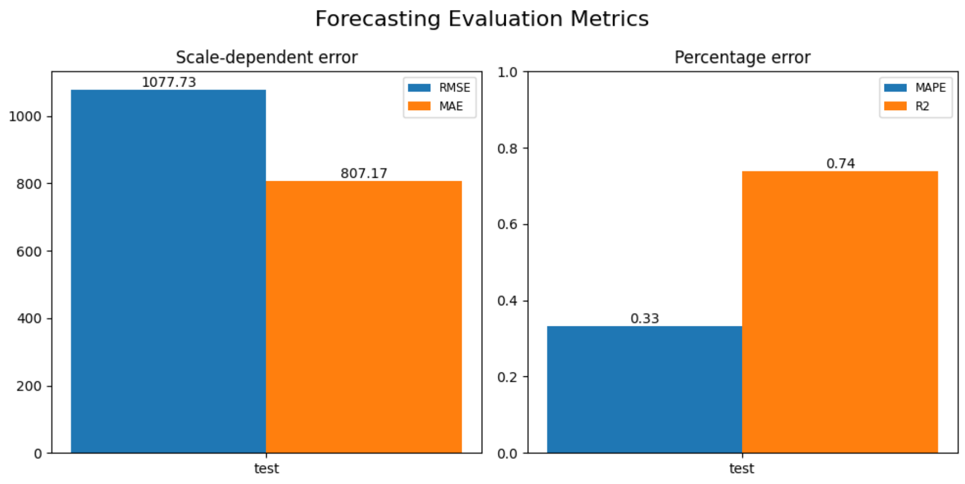


Figure 14: LSTM - First - Forecasting Evaluation Metrics.

4.1.4 Feature Importance and Selection

The next step in model refinement was to assess feature importance and perform feature selection. The goal was to identify the most influential features for traffic volume prediction and potentially improve model performance by reducing noise from less relevant features.

1. **Feature Importance Analysis:** To determine the relative importance of each feature, a permutation-based approach was applied. The process involved the following steps:

   - The baseline prediction error was calculated using the test dataset.
   - For each feature, the corresponding values in the test dataset were shuffled, and the prediction error was recomputed.
   - The importance of the feature was quantified as the absolute difference between the baseline error and the shuffled error, normalized as a percentage of the most impactful feature.

   The results were visualized in a bar plot showing the features ranked by their relative importance:

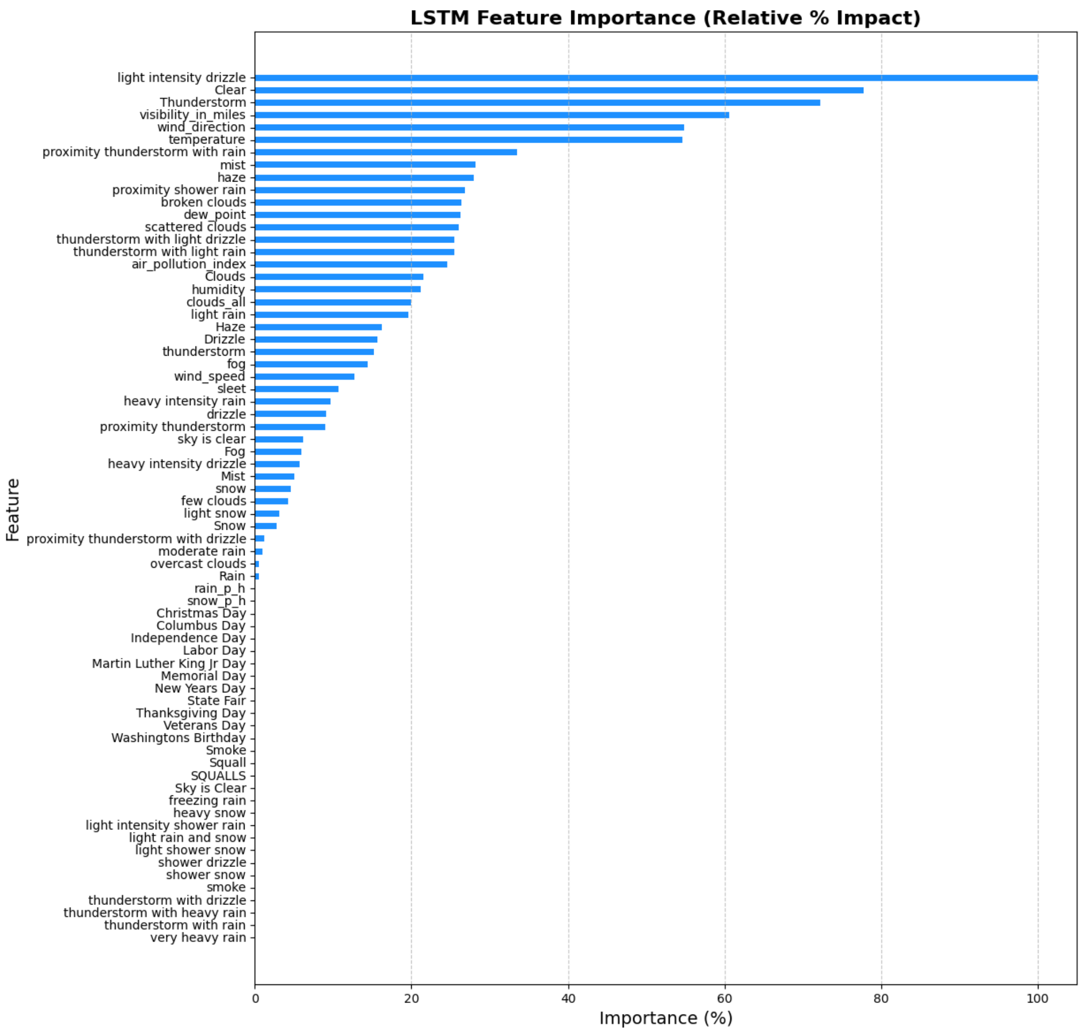


Figure 15: LSTM - Feature Importance (Relative % Impact).

2. **Forward-Stepwise Feature Selection:** Based on the feature importance analysis, features with zero importance were excluded. The remaining features were ordered by their importance, and a forward-stepwise selection approach was implemented:

   - Starting with an empty set of features, one feature was added at a time in the order of importance.
   - For each subset of features, the model was retrained using cross-validation, and its performance was evaluated using the test dataset.
   - The subset of features yielding the best performance (lowest mean squared error) was selected as the optimal feature set.

   The best-performing feature subset identified through this process included: **['light intensity drizzle', 'Clear', 'Thunderstorm', 'visibility_in_miles', 'wind_direction', 'temperature', 'proximity thunderstorm with rain', 'mist', 'haze', 'proximity shower rain', 'broken clouds', 'dew_point', 'scattered clouds', 'thunderstorm with light drizzle', 'thunderstorm with light rain', 'air_pollution_index']**

   This subset of features was used for subsequent modeling experiments to ensure optimal predictive performance while maintaining model interpretability.

4.1.5 Final Model Training with Selected Features

The final model was trained using the best-selected features. The best-performing model from the feature selection step was loaded and evaluated on the test set. The results were visualized in a plot comparing the true and predicted traffic volumes over the last 24 hours:

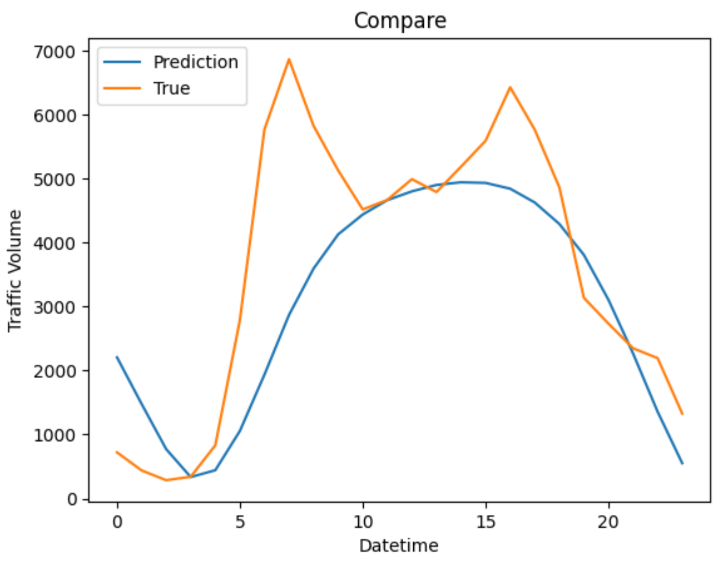


Figure 16: LSTM - Final Model Prediction vs. True Traffic Volume.

Additionally, the final forecasting evaluation metrics were computed and presented:

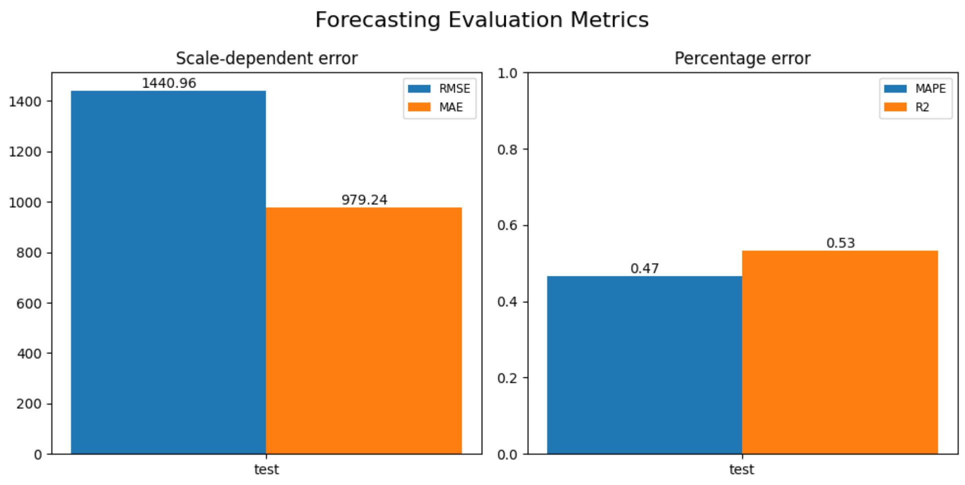


Figure 17: LSTM - Final Forecasting Evaluation Metrics.

This final step ensured that the best model was used with the most relevant features. However, the performance did not improve as expected, indicating that feature selection might have removed useful information or that the model struggled to generalize with the reduced feature set.

4.2 Gated Recurrent Unit (GRU)

4.2.1 Introduction to GRU

Gated Recurrent Unit (GRU) networks are a simplified variant of LSTMs that use gating mechanisms to capture dependencies in sequential data while reducing computational complexity. GRUs do not have a separate cell state like LSTMs but instead use hidden states to manage long-term dependencies. Given their efficiency and ability to model temporal patterns, GRUs were explored as an alternative to LSTMs for traffic volume prediction.

4.2.2 Hyperparameter Tuning

Bayesian Optimization was employed for hyperparameter tuning. This probabilistic approach builds a surrogate model to approximate the objective function and efficiently searches the hyperparameter space, selecting promising configurations based on prior evaluations. The following parameters were tested:

- **GRU Units:** [32, 64, 96, 128, 160, 192, 224, 256]
- **Extra Layer:** True/False
- **GRU Units for Second Layer:** [16, 32, 48, 64, 80, 96, 112, 128]
- **Learning Rates:** [0.01, 0.001, 0.0001]

The best configuration identified was:

- **GRU Units:** 192
- **Extra Layer:** True
- **GRU Units for Second Layer:** 16
- **Learning Rates:** 0.0001

4.2.3 Model Evaluation with the Best Configuration

The best GRU model was evaluated using the test dataset, and predictions were generated for the last 24 hours. The results were visualized and assessed using standard forecasting evaluation metrics.

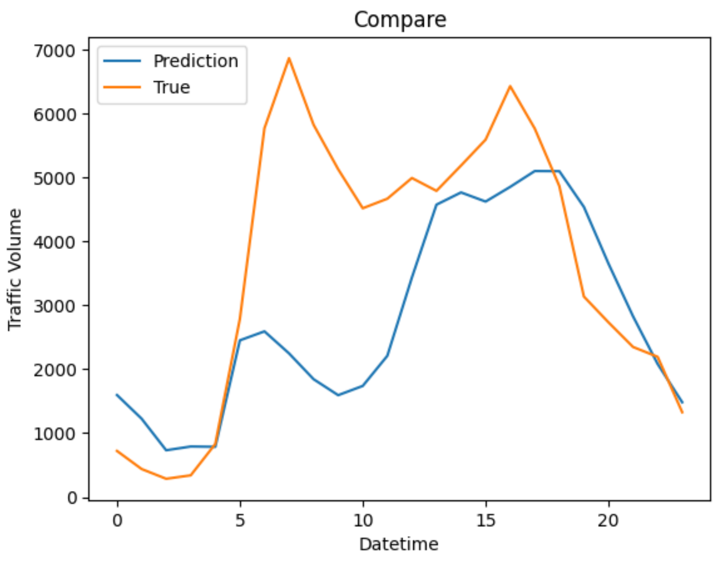


Figure 18: GRU - First - Prediction vs. True Traffic Volume.

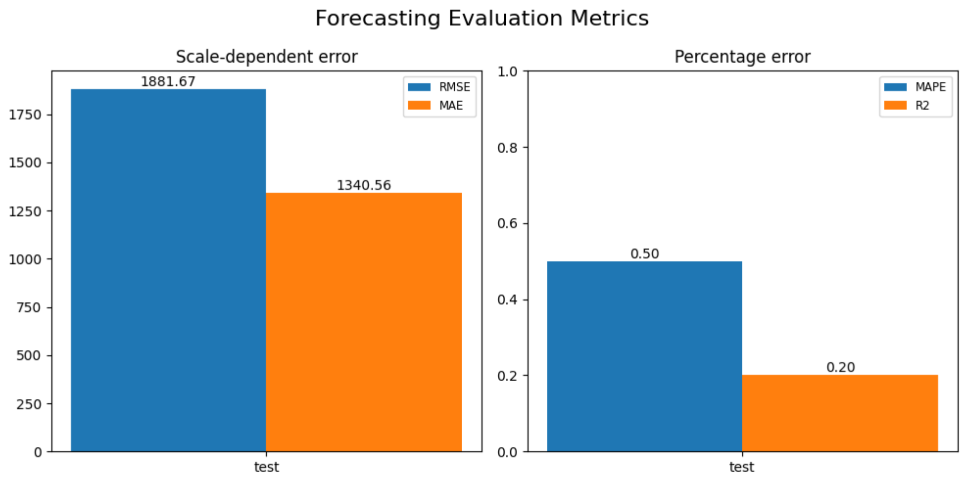


Figure 19: GRU - First - Forecasting Evaluation Metrics.

4.3 Feature Importance and Selection

To identify the most influential features, a permutation-based feature importance analysis was conducted. Each feature was shuffled independently while keeping all others unchanged, and the resulting increase in prediction error was recorded. The features were then ranked based on their relative impact on model performance.

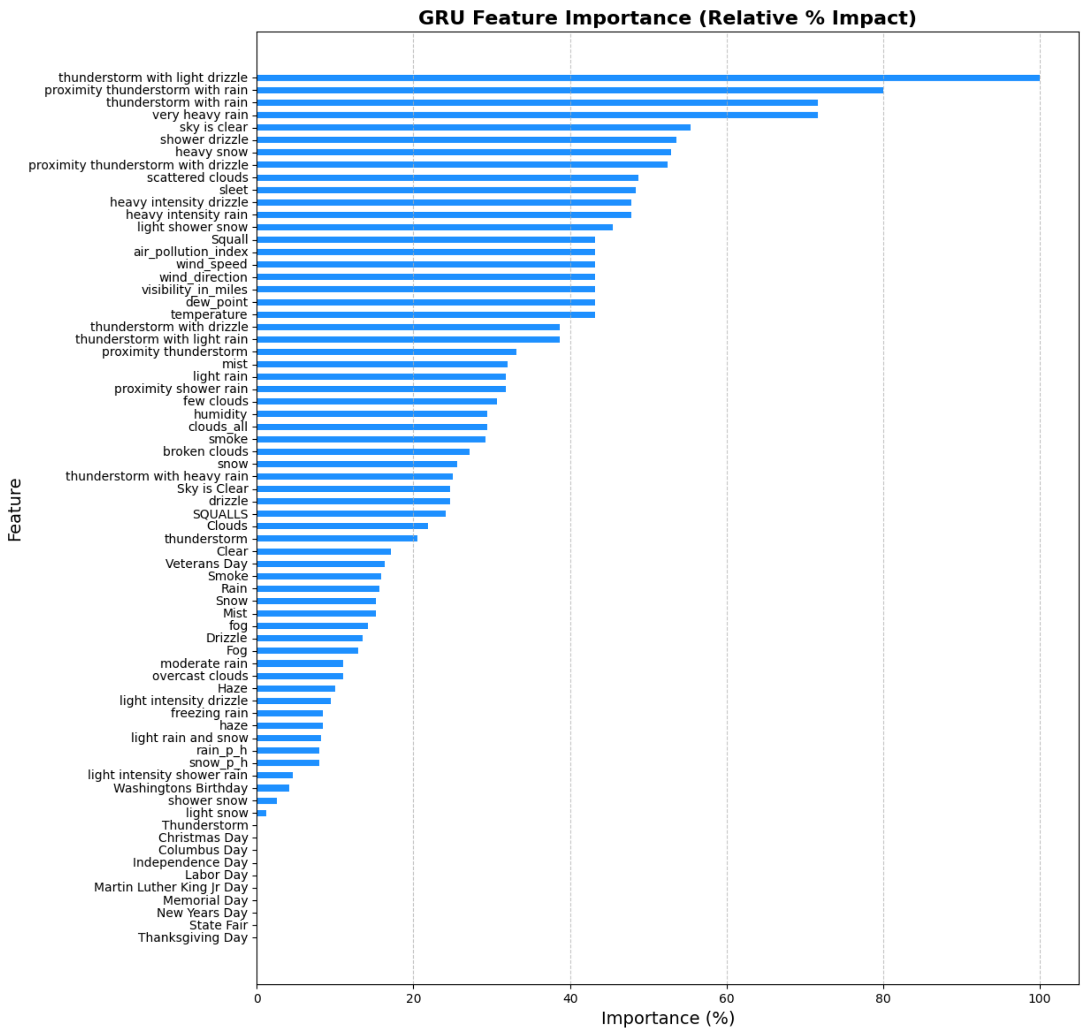


Figure 20: GRU - Feature Importance (Relative % Impact).

To reduce computational complexity, only features with an importance score above 50% were retained for further analysis.

Using the most important features as a base, a forward-stepwise feature selection process was performed. Features were incrementally added based on their importance ranking, and the model was retrained using cross-validation to identify the subset that yielded the best performance.

The optimal feature subset identified through this process consisted of: **['thunderstorm with light drizzle', 'proximity thunderstorm with rain', 'thunderstorm with rain', 'very heavy rain', 'sky is clear', 'shower drizzle']**

4.3.1 Final Model Training with Selected Features

Using the best-selected features, the GRU model was trained again following the same approach as before. However, the final results indicated a decline in performance compared to the previous experiments, suggesting that the GRU model struggled to improve predictions despite feature selection efforts.

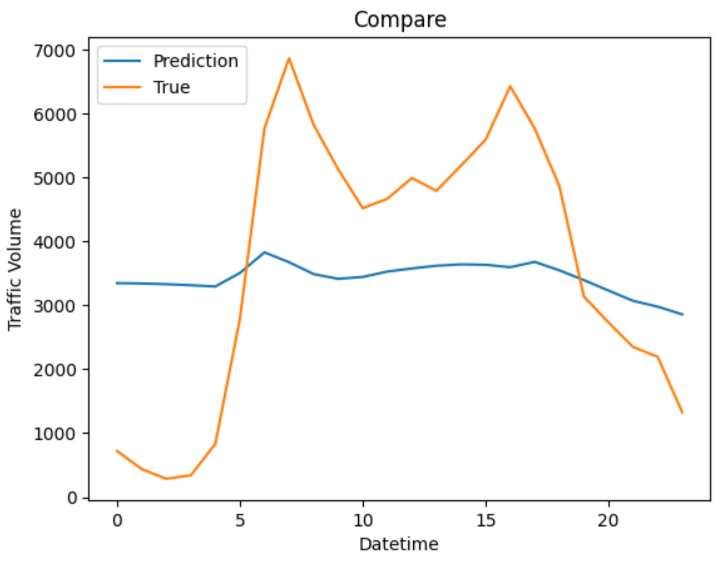


Figure 21: GRU - Final Model Prediction vs. True Traffic Volume.

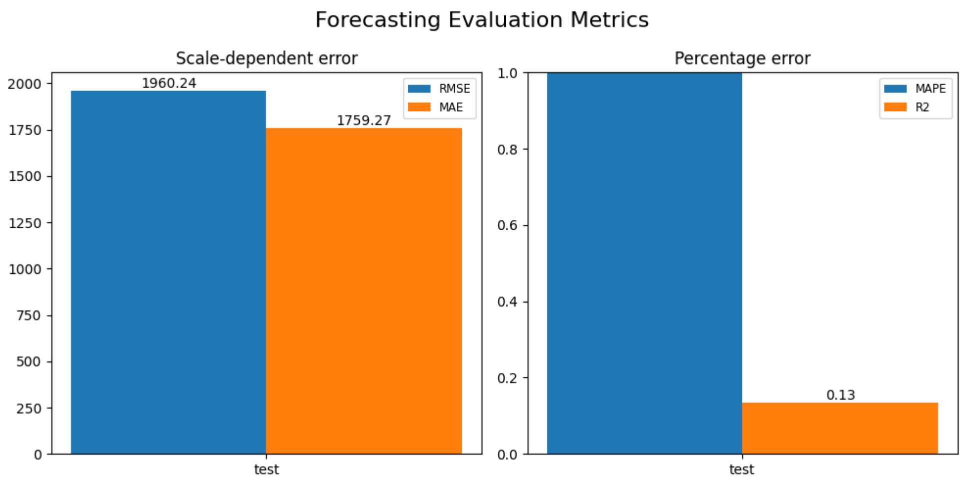


Figure 22: GRU - Final Forecasting Evaluation Metrics.

4.4 Temporal Convolutional Network (TCN)

4.4.1 Introduction to TCN

Temporal Convolutional Networks (TCNs) are a type of deep learning model designed for sequential data processing. Unlike recurrent neural networks (RNNs) such as LSTMs and GRUs, TCNs leverage causal convolutions with dilation, allowing them to capture long-range dependencies efficiently without suffering from vanishing gradient issues. This makes TCNs an attractive option for time-series forecasting tasks, including traffic volume prediction.

4.4.2 Hyperparameter Tuning

To optimize the performance of the TCN model, Bayesian Optimization was employed for hyperparameter tuning. This method efficiently searches the hyperparameter space by building a probabilistic model to approximate the objective function. The following hyperparameters were considered:

- **Number of Layers:** [1, 2, 3, 4]
- **Filters per Layer:** [16, 32, 48, 64, 80, 96, 112, 128]
- **Kernel Sizes:** [2, 3, 5, 7]
- **Dropout Rate:** [0.2, 0.3, 0.4, 0.5]
- **Activation Functions:** [’relu’, ’tanh’, ’sigmoid’, ’swish’]
- **Learning Rates:** [1e-5, 1e-4, 1e-3, 1e-2]

The best hyperparameters identified were:

- **Number of Layers:** 1
- **Filters:** 112
- **Kernel Size:** 2
- **Dropout Rate:** 0.3
- **Activation Function:** ’sigmoid’
- **Learning Rate:** 0.00086

4.4.3 Model Evaluation with the Best Configuration

The best TCN model was evaluated using the test dataset. However, the results were significantly worse than expected. The coefficient of determination (R^2) was found to be -0.035, indicating that the model performed worse than a simple mean-based prediction. The model’s poor performance was further confirmed through visualization, where the predicted values deviated substantially from the true traffic volume values.

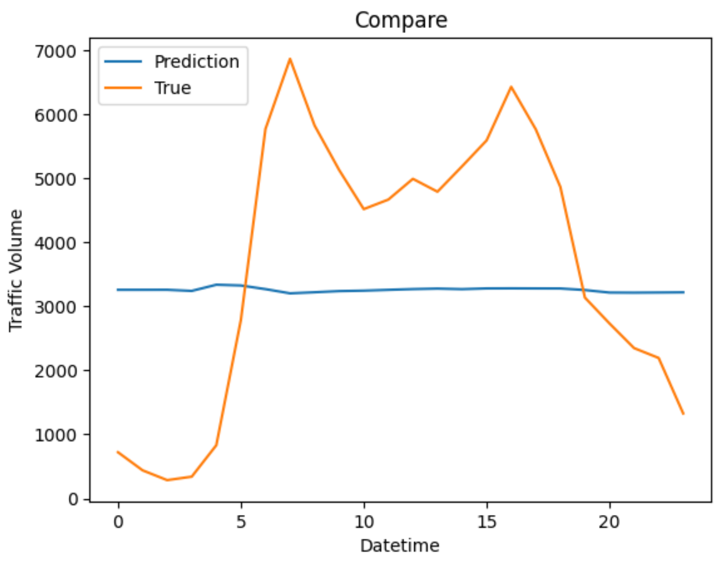


Figure 23: TCN - First - Prediction vs. True Traffic Volume.

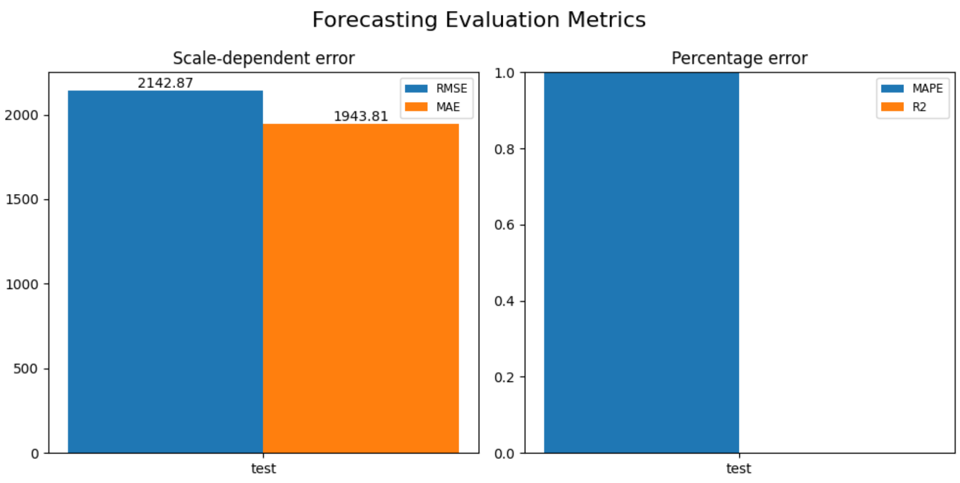


Figure 24: TCN - First - Forecasting Evaluation Metrics.

4.4.4 Feature Importance and Selection

To analyze the contribution of different input features, a feature importance study was conducted. The methodology involved permuting each feature independently and measuring the change in prediction error. The most influential features were identified based on their impact on the model's performance.

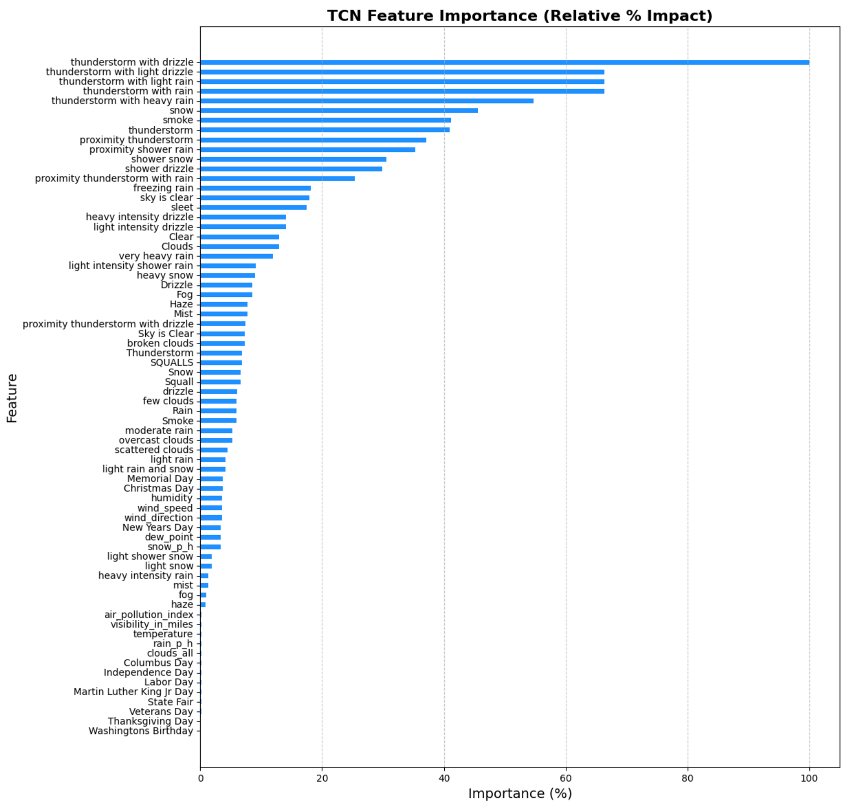


Figure 25: TCN - Feature Importance (Relative % Impact).

To improve model performance, a forward stepwise feature selection approach was applied using the best hyperparameters obtained earlier. This process iteratively added features in order of importance and evaluated their impact on model accuracy.

The best subset of features selected for optimal performance included:

- Various thunderstorm-related conditions (e.g., thunderstorm with drizzle, proximity thunderstorm, heavy intensity rain)
- Weather descriptors such as fog, mist, haze, and air pollution index
- Temperature, humidity, wind speed, and wind direction
- Cloud coverage and visibility
- Public holidays such as Memorial Day, Christmas Day, and New Year's Day

4.4.5 Final Model Training with Selected Features

After selecting the optimal feature subset, the final TCN model was trained again using these features. The best hyperparameters remained the same, and the model was trained following the same procedure as before.

Upon evaluation, the R^2 value improved slightly, reaching 0.01. While this indicates a small gain in predictive performance, the model still struggles to capture significant variance in traffic volume.

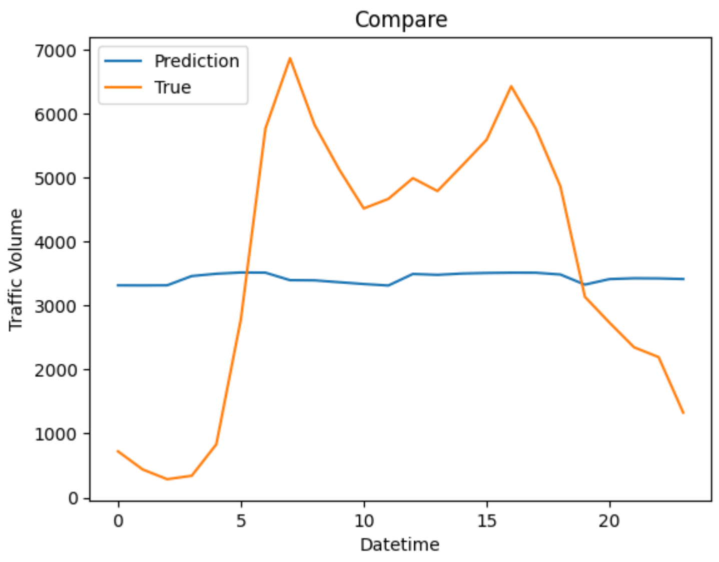


Figure 26: TCN - Final Model Prediction vs. True Traffic Volume.

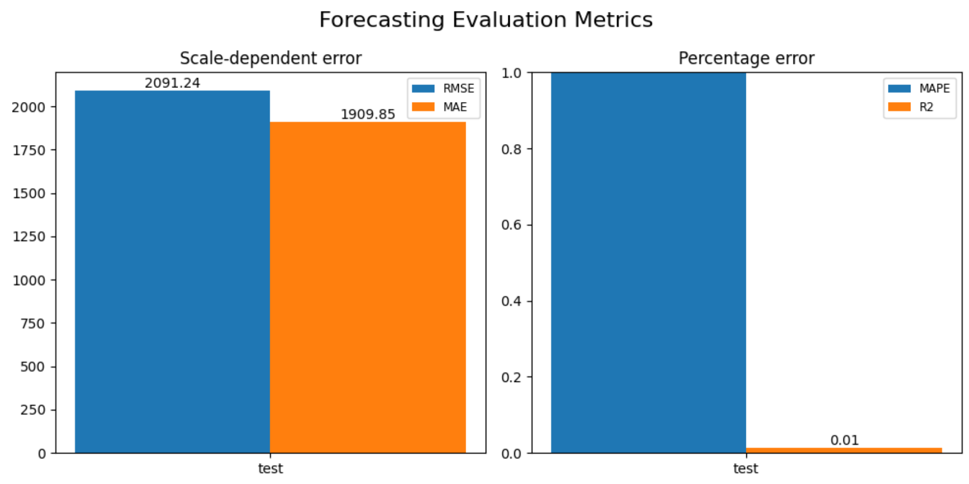


Figure 27: TCN - Final Forecasting Evaluation Metrics.

4.5 Extreme Gradient Boosting (XGBoost)

4.5.1 Introduction to XGBoost

Extreme Gradient Boosting (XGBoost) is a highly efficient and scalable machine learning algorithm based on gradient boosting. It has been widely used in various predictive modeling tasks due to its ability to handle missing values, prevent overfitting, and efficiently process large datasets. Unlike deep learning models like TCN, XGBoost is a tree-based ensemble method that can capture non-linear relationships in the data effectively.

4.5.2 Hyperparameter Tuning

To optimize the performance of the XGBoost model, a hyperparameter tuning process was conducted using GridSearchCV. The following hyperparameters were considered:

- **Number of Estimators:** [50, 100, 200]
- **Max Depth:** [3, 6]
- **Learning Rate:** [0.01, 0.1, 0.3]
- **Subsample:** [0.7, 1.0]
- **Colsample by Tree:** [0.7, 1.0]
- **Gamma:** [0, 0.1]
- **Min Child Weight:** [1, 3]

The hyperparameter search was conducted using 3-fold cross-validation with mean squared error (MSE) as the evaluation metric. The best combination of hyperparameters identified was:

- **Number of Estimators:** 200

- **Max Depth:** 6
- **Learning Rate:** 0.1
- **Subsample:** 0.7
- **Colsample by Tree:** 1.0
- **Gamma:** 0.1
- **Min Child Weight:** 3

4.5.3 Model Evaluation with the Best Configuration

The tuned XGBoost model was evaluated using the test dataset. The performance metrics, including R², mean absolute error (MAE), and root mean squared error (RMSE), were computed. Visualizations were generated to compare predicted and actual traffic volumes.

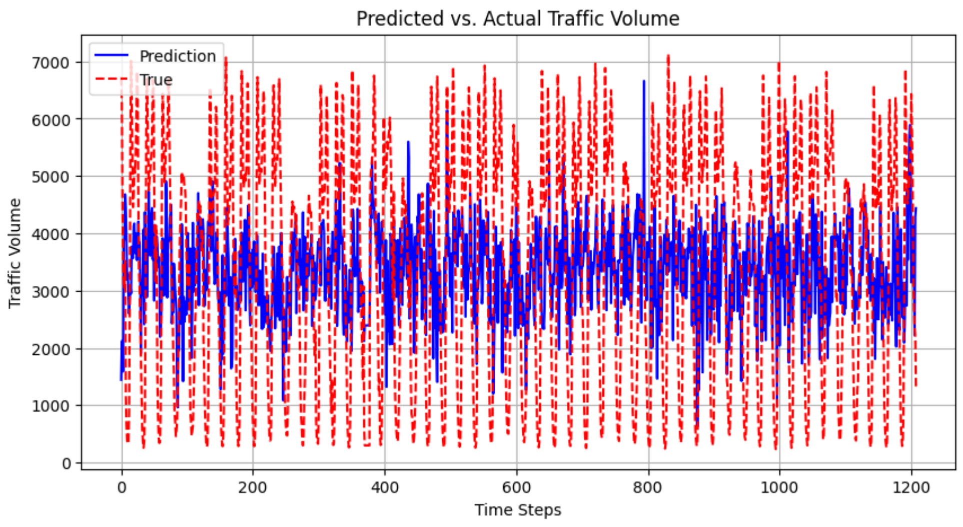


Figure 28: XGBoost - First - Prediction vs. True Traffic Volume.

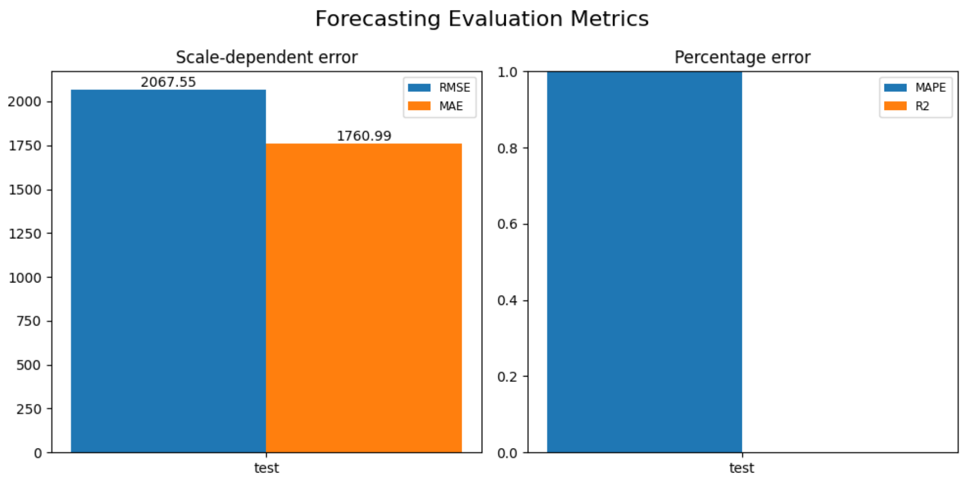


Figure 29: XGBoost - First - Forecasting Evaluation Metrics.

The model yielded an R² value of -0.0309, indicating that it failed to capture the variance in traffic volume data. This negative R² suggests that the model performed worse than a simple mean-based prediction, similar to the previous TCN model. The results imply that further refinements, such as better feature selection or alternative modeling approaches, may be necessary to improve predictive performance.

4.5.4 Feature Importance and Selection

To gain insights into the contribution of different input features, a feature importance analysis was conducted. XGBoost provides a built-in mechanism to assess feature importance based on how often a feature is used in the decision trees and its impact on reducing the error.

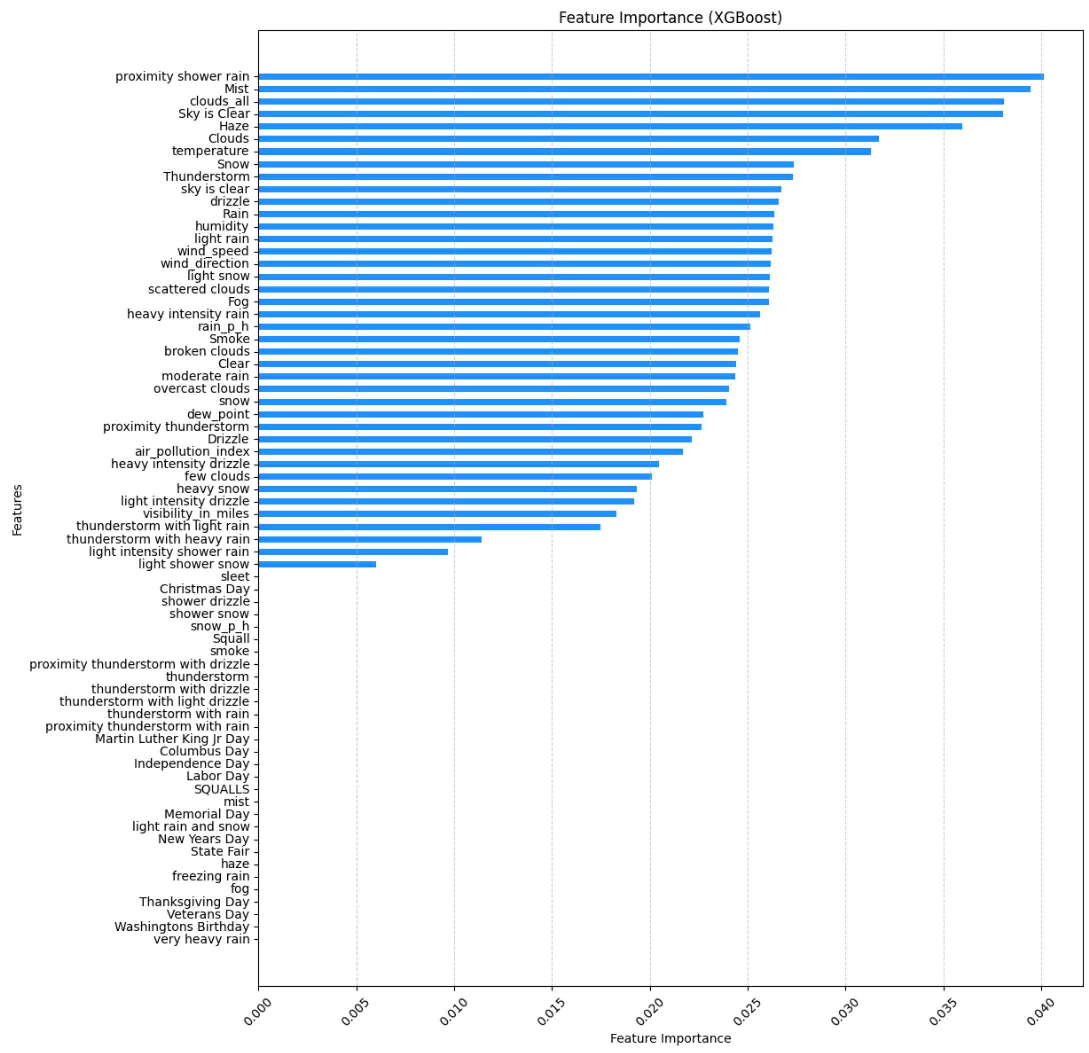


Figure 30: XGBoost - Feature Importance (Relative % Impact).

A forward stepwise feature selection approach was applied to enhance model efficiency and accuracy. This method involved iteratively adding features in order of importance and evaluating their impact on model performance. The best subset of features selected for optimal performance included:

- Proximity Shower Rain

- Mist
- Clouds All
- Sky is Clear
- Haze
- Clouds

4.5.5 Final Model Training with Selected Features

With the optimized hyperparameters and selected features, the final XGBoost model was trained. The model was evaluated on the test dataset to assess its final predictive performance.

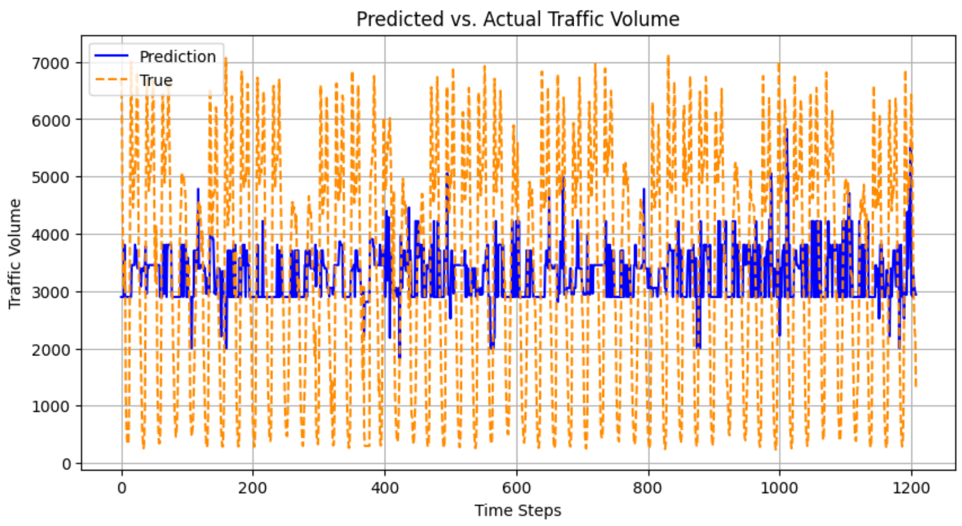


Figure 31: XGBoost - Final Model Prediction vs. True Traffic Volume.

The final model achieved an R² value of 0.05, which is a slight improvement over the initial model's performance (-0.0309). However, this value still indicates a weak correlation between the predicted and actual traffic volume values. While feature selection helped in improving the model, the results suggest that XGBoost may not be the most effective approach for this specific forecasting task.

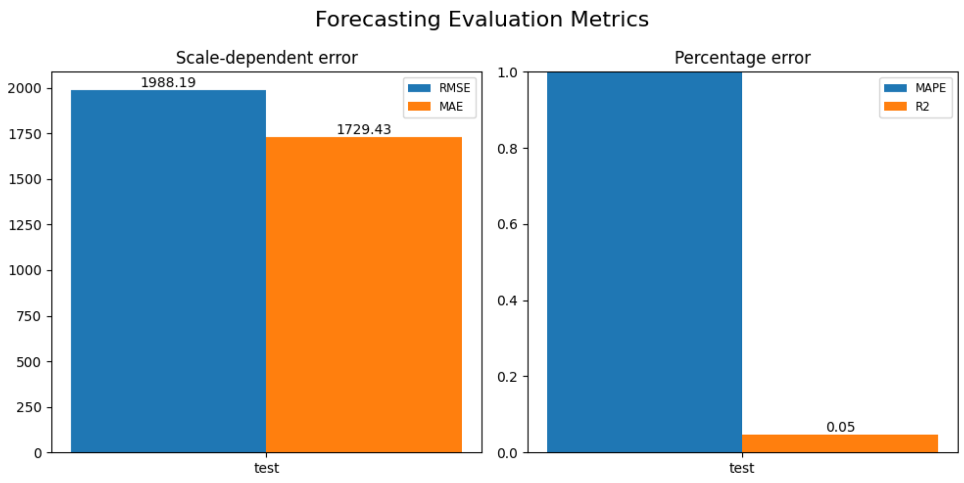


Figure 32: XGBoost - Final Forecasting Evaluation Metrics.

5. Results Discussion

5.1 Best Performing Model Selection

The predictive performance of the various models used in this study was evaluated based on key metrics, including RMSE, MAE, MAPE, and R². The results are summarized in the table below:

Table 1: Best Performance Metrics of each Model

Model	RMSE	MAE	MAPE	R²
Long Short-Term Memory (LSTM)	1077.73	807.17	0.33	0.74
Gated Recurrent Unit (GRU)	1881.67	1340.56	0.50	0.20
Temporal Convolutional Network (TCN)	2091.24	1909.85	>1	0.01
Extreme Gradient Boosting (XGBoost)	1988.19	1729.43	>1	0.05

Based on these results, the LSTM model emerged as the best-performing model, achieving the highest R² value (0.74) and the lowest RMSE and MAE scores. The other models demonstrated significantly weaker performance, with GRU, TCN, and XGBoost struggling to make accurate predictions.

5.2 Structural Analysis of the Models

Poorly Performing Models

- **Gated Recurrent Unit (GRU):** GRU is a recurrent neural network (RNN) variant designed for sequential data, but it appears to have struggled to capture the temporal dependencies effectively in this dataset. The relatively poor performance (R² = 0.20) suggests that the dataset lacks the structured sequential patterns that GRU models typically excel at learning.
- **Temporal Convolutional Network (TCN):** TCN relies on causal convolutions to model sequence dependencies. However, with an R² value close to zero, it is evident that the TCN model was ineffective at detecting any meaningful trends in the traffic data. This indicates that convolutional structures, which excel in image-based tasks, might not be well suited for this particular time-series dataset.
- **Extreme Gradient Boosting (XGBoost):** XGBoost, a powerful tree-based model, generally excels in structured tabular data but may struggle with complex temporal relationships. With an R² of 0.05, it barely provided an improvement over a naïve mean prediction, highlighting its limitations in this context.

The Best Performing Model – LSTM

The **LSTM model** outperformed all others due to its ability to retain long-term dependencies in sequential data. Unlike GRU, TCN, and XGBoost, LSTM effectively modeled the inherent temporal trends present in the traffic dataset. However, despite its superiority, an R² value of 0.74 is still far from ideal for a robust forecasting model.

5.3 The Role of the Dataset in Poor Model Performance

While model selection plays a critical role in predictive accuracy, **the dataset itself is likely the main culprit behind the poor overall performance of the models**. Several issues might have contributed to these suboptimal results:

- **Feature Quality and Relevance:** The dataset contained a range of weather-related features that may not have had a strong direct correlation with traffic volume. Features such as "proximity shower rain" and "haze" were identified as important, yet their practical influence on traffic patterns remains questionable.
- **Data Noise and Inconsistencies:** High levels of noise, missing values, and inconsistencies in the dataset could have hindered the models' ability to learn meaningful patterns.
- **Temporal Resolution Issues:** If the dataset lacks sufficient temporal granularity, certain trends and fluctuations in traffic volume may not have been adequately captured by the models.
- **Exogenous Factors Not Considered:** Many real-world variables influencing traffic volume, such as road closures and special events, were likely not accounted for in the dataset. This absence of critical external factors could have prevented the models from making accurate predictions.

While LSTM proved to be the best-performing model, **the dataset itself posed significant challenges, limiting the potential accuracy of all models**.

6. Results Discussion

This study aimed to develop and evaluate various machine learning models for traffic volume prediction using a dataset with multiple environmental and temporal features. Four models (LSTM, GRU, TCN, and XGBoost) were implemented, fine-tuned, and assessed using key performance metrics. Among these, **the LSTM model demonstrated the best performance**, achieving an R^2 of 0.74, while the other models struggled to generalize well, with significantly lower R^2 values. However, even the best-performing model leaves room for improvement, as it does not achieve the level of accuracy expected in practical applications.

The analysis highlighted that **the dataset itself played a crucial role in limiting model performance**. The presence of weakly relevant features, noisy and inconsistent data, and the exclusion of critical exogenous factors (such as traffic incidents, road infrastructure, and socio-economic events) likely contributed to the suboptimal results. Weather-related variables, while showing some correlation with traffic volume, did not provide sufficient predictive power to build a highly accurate forecasting model.

Future research should focus on improving traffic prediction accuracy through dataset enrichment by incorporating additional features such as traffic flow from connected sensors, roadwork reports, public transport schedules, and special event data. A more refined feature selection process using automated methods like recursive feature elimination (RFE) or dimensionality reduction techniques could help eliminate noise. Hybrid models, including LSTMs with attention mechanisms, Transformer-based architectures, or ensemble approaches, may further enhance predictive performance. Addressing missing values, outliers, and inconsistencies in data collection would improve model generalization. Additionally, developing real-time traffic prediction systems that integrate streaming data with batch learning could provide practical applications. While this study offered valuable insights into machine learning for traffic forecasting, it also highlighted challenges, particularly in dataset quality. Future efforts should aim to overcome these limitations for more reliable traffic prediction models.