Master of Science in Data Analytics

An Intelligent Traffic Congestion Prediction System using Machine Learning

Nupur Pathak, Sree Divya Cheerla, Vani Bhat

Department of Applied Data Science, San Jose State University

DATA 240: Data Mining and Analytics

Project Report

Prof. Shayan Shams

May 18, 2023

1. Introduction

1.1 Problem

Traffic congestion is a complex and challenging problem that requires meticulous planning and management to lessen its effects which include economic losses, increased fuel consumption, and negative impacts on the environment. As the world's population grows day by day, there is sharp rise in the number of individuals who must travel to work, school, and other activities. Rapid urbanization with more people moving from rural to urban areas adds more strain on transportation management. In addition, factors such as insufficient infrastructure, ineffective traffic management, and poor driving behavior also contribute to an increase in traffic congestion.

Traffic congestion can be caused due to a variety of factors, including road network design, traffic flow, driver behavior, and environmental factors such as weather conditions. Traditional methods such as statistical models have limitations in terms of accuracy and scalability as these models may not consider the dynamic nature of traffic flow and the impact of changing conditions, such as weather and accidents. This project aims to address the problem of traffic congestion by developing an intelligent traffic congestion prediction system using machine learning algorithms.

1.2 Motivation

Traffic congestion is one of the major problems that affects people’s quality of life in urban areas as it leads to longer travel times, increased fuel consumption, and air pollution. In addition, delays in the delivery of goods and services which are caused by traffic congestion can cause economic losses which is why it is important to develop effective solutions to mitigate its effects. By predicting traffic congestion in real-time, we can provide drivers with alternate routes and reduce the impact of traffic congestion on the environment and the economy.

The primary motivation behind this project is to develop an intelligent traffic congestion prediction system which helps to alleviate the negative impacts of traffic congestion. With real-time traffic, the drivers can be provided with alternate routes reducing the travel times and decrease in fuel consumption and minimizing the environmental impacts. Moreover, accurate traffic predictions can enable city planners to make informed decisions about infrastructure development and traffic management.

1.3 Background

Traditionally, for traffic congestion prediction, statistical methods such as time-series analysis and regression models are used which have limitations in terms of accuracy and scalability. With the advent of machine learning algorithms, prediction of traffic congestion can be done more accurately and in real-time.

Machine learning algorithms can be used to learn the patterns from historical traffic data and use them to make predictions about future traffic conditions. Some of the popularly used machine learning algorithms for traffic congestion prediction are neural networks, decision trees, and support vector machines. Our proposed system will use historical traffic data and road network information for training the machine learning algorithm. Once the model is trained, it will then be used to make real-time predictions about traffic congestion.

1.4 Literature Review

Research Paper	Authors	Business Objective	Models Used	Evaluation
Traffic Congestion Prediction Using Machine Learning Techniques	Yasir et al.(2022)	Proposed a prediction model for the traffic congestion that can predict congestion based on day, time and several weather data (e.g., temperature, humidity).	SVR (Support Vector Regressor)	RMSE
Gradient Boosting Approach for Traffic Flow Prediction using CatBoost	Singh et at. (2021)	Proposed an approach which considers important factors such as no. of intersections on the street, no. of commercial places near the street, and structure of the street.	BPNN, XGBoost, CatBoost, CatBoost with hyperparamet er tuning	Accuracy Precision F1 score Recall
Prediction of Road Traffic Congestion Based on Random Forest	Liu1 et al.(2017)	Weather conditions, time period, special conditions of road, road quality and holiday are used as model input variables to establish road traffic forecasting model	Random Forest	Accuracy
Short term traffic flow prediction based on combination model of xgboost-lightgbm	Mei et al.(2018)	Proposed a combined prediction model where xgboost and lightgbm are constructed individually and later merged to generate the final model.	Xgboost and lightgbm	MAPE (mean absolute percentage error)

By using both boosting and bagging strategies to address the issue of traffic congestion prediction, we distinguish our study's methodology from the literature review that has already been conducted. While previous authors in the field have traditionally used either boosting or bagging methods separately, we understand the advantages of both strategies and compare them in our analysis. This innovative pairing enables us to identify various patterns and fluctuations in traffic data, improving the precision and robustness of our forecasts.

2. Methodology

2.1 Data Collection

Traffic intersection congestion dataset consists of aggregated trip logging metrics from commercial vehicles. The data for the traffic information is captured at various intersection points at different time stamps. It contains 856k observations aggregating stopped vehicle information and intersection wait times. The information is captured at intersections in 4 major US cities: Atlanta, Boston, Chicago & Philadelphia. It comprises historical traffic data, such as entry street, exit street, intersection details, direction driven at the intersection, the time required to travel the street, month and hour of the day, and weekend indicator. The weather information on temperature and rainfall of each city by month is collected from NOAA National Climatic Data Center (NCDC) and joined with the observations.

After the collected data, an exploratory data analysis (EDA) was performed using a data quality report to identify various characteristics of the dataset, including the count, cardinality, missing values, and outliers for both categorical and continuous features.

2.2 Data Pre-Processing

The key steps involved in the data preprocessing are data cleaning, feature engineering, data scaling, and data splitting. In data cleaning, duplicate observations are checked and removed. The features EntryStreetName and ExitStreetName had missing values and those observations have been removed.

In feature engineering, relevant features from the dataset can potentially provide meaningful insights. Following feature transformations have been conducted:

- Mapping directions: EntryHeading and ExitHeading are mapped with corresponding directions (in radians) for better compatibility with the machine learning models.
- Encoding road types: A dictionary is used to encode street names into numerical values, with higher values indicating a higher road type. This is done to provide an ordered representation of street names that can be used by the machine learning models.
- Difference in heading: diffHeading column is created by subtracting ExitHeading from EntryHeading to give the difference in the heading of the car. This helps to capture the angular change in the direction of the car, which can be useful for predicting traffic congestion.
- Same street check: A binary column same_street_exact is created which takes value 1 if the EntryStreetName and ExitStreetName are the same, and 0 otherwise. This is done to capture cases where cars take a U-turn or change lanes to continue on the same road.
- Label Encoding Intersection: IntersectionId and City columns are combined to form a unique identifier for each intersection. Label encoding is applied to encode each intersection as a numerical value for better compatibility with the machine learning models.
- Temperature of each city by month: Monthly average temperature of each city is added as a feature by mapping the city-month variable to its corresponding average monthly temperature. This can help to capture the effect of weather on traffic congestion.
- Rainfall of each city by month: Monthly average rainfall of each city is added as a feature by mapping the city-month variable to its corresponding average monthly rainfall. This can help to capture the effect of weather on traffic congestion.

In data scaling, data normalization is performed, and feature values are scaled to [0, 1]). For categorical features, label encoding is performed.

2.3 Modeling

Based on the literature survey, four models have been selected. These are Random Forest, CatBoost, XGBoost, and LightGBM.

Random Forest: Random Forest is one of the most used ensemble machine learning algorithms which combines the predictions of multiple decision trees to make more accurate predictions. It is popularly used for classification and regression tasks, including traffic flow prediction. The algorithm is built by using a technique called bagging which is an ensemble of decision trees where each tree is trained on a random subset of the training data.

The model builds multiple decision trees by using different subsets of the training data so that each decision tree is constructed by selecting features at each node randomly and splitting the data on the best split based on some criterion (e.g., Gini impurity or Information gain). Once the decision trees are constructed, predictions are made for each individual tree. For classification tasks, from each tree a “vote” is generated for a class label, and the class with the majority of votes is selected as the final prediction. For regression tasks, like traffic flow prediction, the average of individual tree predictions is considered to obtain the final prediction.

For traffic flow prediction, the model considers various other features related to traffic, such as traffic flow history, weather conditions, time, day of the week. During training, the model would learn the patterns and relationships between the input features such as historical traffic flow, weather conditions and the target variable from the training data. The random feature selection and subset sampling is performed during training which helps in capturing different aspects of traffic flow patterns.

CatBoost: CatBoost is a gradient boosting algorithm that is particularly beneficial for traffic congestion analysis due to its unique features and capabilities. CatBoost is designed to handle categorical variables well. It makes the addition of categorical features easier and requires less complex feature engineering by using Ordered Target Encoding. CatBoost uses robust methods to manage noisy or missing data. Robust to Noisy Data. It can automatically manage outliers and missing values, lessening the workload associated with data cleaning and preparation.

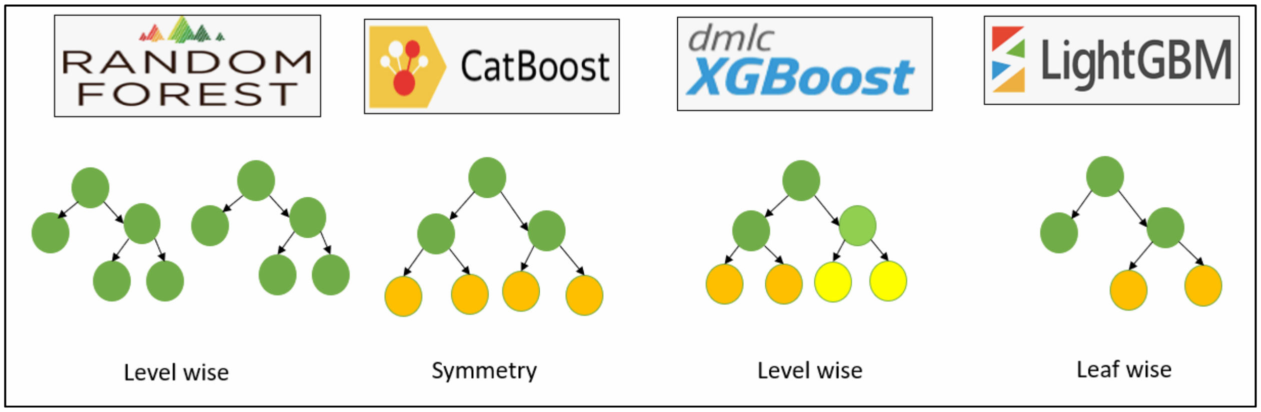
In addition, CatBoost sequentially trains an ensemble of weak models to enhance performance. It successfully handles numerical and categorical information, capturing intricate relationships and patterns in the data. It optimizes the boosting process and minimizes a variety of loss functions to produce excellent predictive performance. It is appropriate for use in real-world applications since it can handle big data sets with plenty of features and observations. Cross-validation functionality is included in CatBoost, allowing for trustworthy model evaluation and hyperparameter adjustment. To create reliable and accurate models, you must have this feature.

XGBoost: Extreme Gradient Boosting (XGBoost) is an advanced ensemble learning algorithm widely known for its excellent performance in various machine learning tasks. It is an optimized implementation of the gradient boosting framework that can handle complex, high-dimensional datasets. XGBoost uses a boosting framework, where multiple decision trees are sequentially trained to correct the errors made by the previous models. During training, the model learns to minimize the loss function using gradient optimization techniques. By calculating the gradient of the loss function based on the difference between predicted values and actual values, it updates the model's parameters (weights).

XGBoost measures feature importance, considering the relative contribution of each feature to the model's predictions which helps to identify the most important factors in traffic flow prediction. For traffic flow prediction, the model iteratively builds decision trees, by adjusting the weights of the observations based on the errors made by previous decision trees. By incorporating gradient optimization techniques which minimize the loss function and updates the model parameters, the model gets trained. In addition, Regularization techniques like L1, L2 are applied to prevent the model from overfitting and to enhance generalization.

LightGBM: LightGBM is a high-performance gradient boosting framework that stands for Light Gradient Boosting Machine. It is a gradient-boosting framework that specifically focuses on optimizing and enhancing the training and inference processes of decision tree-based models. The base classifier is fixed to decision trees here. LightGBM builds decision trees in a leaf-wise manner, which means that it grows the tree by splitting the leaf with the maximum delta loss and only one leaf can split at a time. This approach tends to improve training efficiency and can lead to better accuracy compared to other traditional depth-wise tree growth methods. LightGBM employs a unique approach known as Gradient-based One-Side Sampling (GOSS) to select and prioritize the most informative data samples during the boosting process, resulting in faster training speeds and reduced memory consumption. It also utilizes a histogram-based algorithm to discretize continuous features into bins, enabling efficient computations and reducing the impact of outliers. In summary, LightGBM's efficiency, handling of categorical features, accuracy, interpretability, memory efficiency, and flexibility make it a well-suited framework for traffic congestion prediction tasks. Its ability to handle large datasets, capture complex relationships, and provide accurate predictions can aid in better understanding traffic patterns, optimizing traffic management strategies, and improving overall transportation systems.

Following figure shows the four ensemble techniques and their decision tree:



2.4 Training

The data is split into training and testing subsets in the ratio of 80:20. The training set is used to build the model, while the testing set is used to evaluate its performance. Ensure that the data split is representative and random to avoid introducing bias in subsequent analysis or modeling tasks. For better estimate of the model’s performance, k-fold cross validation is implemented with k=5.

K- Fold cross validation is an essential technique used for evaluating the performance of the models. It involves dividing the entire training dataset into k - folds or subsets and training the model for k number of interactions so that each fold is used as a testing set once. The average performance of all the k-folds is considered as the final estimate. This technique helps to reduce overfitting and provides a better estimate of the model’s performance. In addition, it also helps to identify the presence of bias or variance which affects the model’s performance.

3. Results

Our project focuses on tackling traffic congestion prediction as a regression problem. We aim to predict six target features: TotalTimeStopped_p20, TotalTimeStopped_p50, TotalTimeStopped_p80, DistanceToFirstStop_p20, DistanceToFirstStop_p50, and DistanceToFirstStop_p80 to predict the traffic congestion times. To achieve this, we proposed four machine learning algorithms: Random Forest, XGBoost, LightGBM, and CatBoost. By using decision trees as the base classifiers for all models, we can effectively compare and contrast their performance and gain insights into their individual characteristics. We have used RMSE and R2 as the evaluation metrics.

Initially, we conducted experiments by training all four models with their default hyperparameters. Our observations revealed that XGBoost exhibited the highest root mean squared error (RMSE) of approximately 44.9, followed by CatBoost, Random Forest, and LightGBM. However, the R2 scores showed a different order, with Random Forest leading the way with a score of 0.52, followed by LightGBM (0.43), CatBoost (0.36), and XGBoost (0.25). For the given dataset, although XGBoost had higher RMSE, it was able to explain a significant amount of variance in the target variables, as reflected by its R2 score. On the other hand, Random Forest had the lowest RMSE, indicating better accuracy, and achieved the highest R2 score, implying a stronger overall explanatory ability.

Additionally, we measured the training time for each model, and the order from highest to lowest was CatBoost, XGBoost, Random Forest, and LightGBM. Notably, LightGBM demonstrated the fastest training speed among the models as it uses a histogram-based approach.

Traffic congestion is influenced by several factors, including climate conditions such as weather, rainfall, and snowfall. In our study, we aimed to investigate the impact of these factors on congestion prediction by incorporating additional features such as average temperature and average rainfall for each city. Our objective was to determine if these climate details would lead to notable variations in the results. However, upon analyzing the data, we found that the inclusion of climate details did not result in significant changes to the outcomes. The effects on congestion prediction remained relatively unchanged even after considering climate-related variables.

Table 1

	Metrics	**XGBoost**	**CatBoost**	**Random Forest**	**Light GBM**
Before Climate details	RMSE	44.1913	39.54	32.8497	36.66
	R2_Score	0.25	0.36	0.52	0.43
	Time_taken	9 min	13 min	9 min	3.02 min
	Metrics	**XGBoost**	**CatBoost**	**Random Forest**	**Light gbm**
After Climate details	RMSE	44.2038	39.66	32.8607	36.92
	R2_Score	0.25	0.36	0.52	0.42
	Training time	11 min	12 min	11 min	2.93 min

To further enhance the models' performance, we proceeded to fine-tune their hyperparameters. This involved systematically exploring various combinations of hyperparameters and evaluating the models using 5-fold cross-validation to observe for any improvements in the RMSE and R2 score. For each of the models we experimented with the hyperparameters below.

XGBoost: Experimented with hyperparameters like num_leaves, max_depth, learning_rate and n_estimators. For max_depth(maximum depth of a tree) the experimental values were 5,7 and where it was observed that 7 gave better performance comparatively. For n_estimators, the experimental values were 100, 150 and 500, where it was observed that 150 gave better performance. For learning rate, the experimental values were 0.05 and 0.1, where it was observed that 0.1 gave optimal performance.

Random Forest: Experimented with hyperparameters like min_samples_split, learning_rate and n_estimators. For min_samples_split (minimum number of samples required to split an internal node), the experimental values are 3 and 5, where 3 gave better performance. For n_estimators, the experimental values are 100 and 150 from which 150 gave better performance. For learning rate, the experimental values were 0.05 and 0.1, where it was observed that 0.1 gave optimal performance.

CatBoost:Experimented with hyperparameters like min_data_in_leaf, learning_rate and n_estimators. For min_data_in_leaf, we have tested with 5 and 10. We have then fine-tuned it to 10 as better results were achieved. For learning rate, the experimental values were 0.05 and 0.1, where it was observed that 0.1 gave optimal performance. The n_estimators were changed from 100 to 150.

LightGBM: We conducted experiments involving the tuning of hyperparameters such as num_leaves, max_depth, learning_rate, and n_estimators to optimize the performance of our models. For the num_leaves parameter, which determines the maximum number of leaves in a tree, we explored values of 128, 256, and 512. Through our observations, we found that using 512 as the value for num_leaves resulted in better performance when compared to the other options.In the case of the max_depth parameter, which controls the maximum depth of each tree, we experimented with values of 7, 8, and 9. From our analysis, it became evident that setting max_depth to 9 yielded superior results compared to the other choices. When fine-tuning the learning_rate, which determines the step size during the boosting process, we discovered that the default value of 0.1 worked best for our LightGBM model.

Overall, training the models with these hyperparameter experiments and k-fold cross validation contributed to reducing the root mean squared error (RMSE) and increasing the R2 score of our models. By selecting appropriate values for num_leaves, max_depth, learning_rate, and n_estimators we achieved better overall performance, resulting in improved accuracy in predicting traffic congestion. The best values are updated in the below table. We can see that RandomForest gave the best results with RMSE 32.63 and R2 Score 0.52 comparatively.

Table 2

	Metrics	XGBoost	CatBoost	Random Forest	Light GBM
K-fold	**RMSE**	54.66	53.23	32.63	36.94
	R2_Score	0.34	0.35	0.52	0.42
	Training time	~68 min	~46 min	~18 min	~10 min
	Fine-tune Hyperparameters	learning rate = 0.1 n_estimator = 150 max_depth = 7	Learning_rate = 0.1 N_estimators = 150 Max_depth = 6 Min_data_in_leaf = 10	learning rate = 0.1 n_estimator = 150	Learning_rate = 0.1 Num_leaves = 512 max_depth= 9 N_estimators = 150

4. Future Scope

Traffic congestion is a major problem in many urban areas, leading to increased travel times, air pollution, and fuel consumption. Accurately predicting traffic congestion in advance can help develop proactive measures to optimize traffic flow, reroute vehicles, and provide real-time information to commuters, empowering them to make informed decisions. City planners could also use these models to plan and design new roads and transportation systems, considering predicted traffic patterns. Several improvements can be thought of considering the importance of the problem.

- Enhanced Data Integration and Quality: Future improvements can focus on expanding data sources and ensuring data quality, consistency, and coverage across different regions and transportation modes. Integrating real-time data sources like traffic cameras, GPS data, and sensor networks can significantly enhance the accuracy of congestion predictions.
- Advanced Machine Learning Techniques: As machine learning algorithms and models continue to evolve, future improvements can focus on utilizing Deep learning algorithms, ensemble methods, and hybrid models that combine multiple data sources and modelling approaches to enhance the accuracy and robustness of traffic congestion predictions.
- Predictive Analytics for Traffic Management: Integrating predictive analytics tools with traffic prediction systems enables proactive congestion management. By identifying congestion hotspots, predicting peak traffic periods, and optimizing traffic flow in real-time, transportation agencies can implement effective traffic management strategies.
- Integration with sustainable transportation solutions: By integrating predictions with public transportation schedules, bike-sharing programs, and ride-sharing services, congestion can be mitigated by encouraging alternative modes of transportation and reducing the number of private vehicles on the road.

5. References

- Yasir, Rafed Muhammad, et al. *Traffic Congestion Prediction Using Machine Learning Techniques*, https://doi.org/ https://doi.org/10.48550/arXiv.2206.10983.
- Singh, Rajeev, et al. “Gradient Boosting Approach for Traffic Flow Prediction Using CatBoost.” *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)*, 2021, https://doi.org/10.1109/icac353642.2021.9697133.
- Liu, Yunxiang, and Hao Wu. “Prediction of Road Traffic Congestion Based on Random Forest.” *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, 2017, https://doi.org/10.1109/iscid.2017.216.
- Mei, Zhang, et al. “Short-Term Traffic Flow Prediction Based on Combination Model of Xgboost-LIGHTGBM.” *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, 2018, https://doi.org/10.1109/snsp.2018.00069.
- BigQuery-Geotab Intersection Congestion Data: https://www.kaggle.com/competitions/bigquery-geotab-intersection-congestion/data