­­­Traffic congestion prediction using Machine learning algorithm

# Abstract

Congestion in the transportation industry is a major problem that calls for accurate forecasting techniques. In this investigation, we apply machine learning strategies to the problem of predicting when and where traffic congestion will occur. The goal is to enhance approaches to traffic management, including planning, resource allocation, and operations. Using a thorough technique that includes data preprocessing, exploratory data analysis, and the assessment of several machine learning models, the experiment is carried out. This research makes use of a dataset retrieved from the Kaggle platform; it includes information on traffic volumes, times of day, and intersections. The quality and integrity of the data is guaranteed during the preprocessing phase by the removal of any invalid or duplicate information. Temporal features are extracted using feature engineering approaches, allowing for fine-grained study of traffic patterns and fluctuations. The causes and effects of traffic congestion can be better understood by exploratory data analysis, which improves decision-making and resource allocation. Predictions of traffic congestion are made using a variety of machine learning models, such as Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, K Nearest Neighbours, Stacked Model, and Blending Model. Mean squared error, root mean squared error, and mean absolute error are some of the regression metrics used to evaluate the models. The Gradient Boosting Regressor emerges as the most accurate model, demonstrating its superior predictive power in capturing traffic congestion patterns.

# Acknowledgment

Table of Contents

[Abstract 2](#_Toc140497506)

[Acknowledgment 3](#_Toc140497507)

[Chapter 1- Introduction 6](#_Toc140497508)

[1.1 Background 6](#_Toc140497509)

[1.2 Purpose of this study 8](#_Toc140497510)

[1.3 Aim and Objectives 8](#_Toc140497511)

[1.4 Outline of the dissertation 8](#_Toc140497512)

[Chapter 2- Literature review 9](#_Toc140497513)

[2.1 Machine learning for Traffic congestion prediction 10](#_Toc140497514)

[2.2 Multimodal framework for the Traffic congestion prediction 11](#_Toc140497515)

[2.3 Deep learning algorithms for performing the Traffic congestion prediction 12](#_Toc140497516)

[2.5 Research gaps 15](#_Toc140497517)

[2.6 Summary 18](#_Toc140497518)

[Chapter 3- Methodology 18](#_Toc140497519)

[3.1 Data collection 18](#_Toc140497520)

[3.2 Ethical Considerations 19](#_Toc140497521)

[3.3 Requirement Specifications 19](#_Toc140497522)

[3.4 Checking the quality of Data 19](#_Toc140497523)

[3.5 Data Preprocessing 20](#_Toc140497524)

[3.6 Machine learning algorithms 20](#_Toc140497525)

[3.6.1 Linear regression 20](#_Toc140497526)

[3.6.2 Random forest regression 21](#_Toc140497527)

[3.6.3 Gradient boosting regressor 22](#_Toc140497528)

[3.6.4 K nearest neighbour 22](#_Toc140497529)

[3.6.5 Stacking model 23](#_Toc140497530)

[3.6.7 Blending model 24](#_Toc140497531)

[3.7 Hyperparameter tunning 24](#_Toc140497532)

[3.8 Evaluation metrics 25](#_Toc140497533)

[3.9 Summary 25](#_Toc140497534)

[Chapter 4- Implementation, Results and Discussion 25](#_Toc140497535)

[4.1 Implementation and Results 25](#_Toc140497536)

[4.2 Discussion 32](#_Toc140497537)

[Chapter 5- Conclusion and Future work 33](#_Toc140497538)

[5.1 Conclusion 33](#_Toc140497539)

[5.2 Research Contributions 33](#_Toc140497540)

[5.3 Future Work and Developments: 33](#_Toc140497541)

[5.4 Personal Reflections 33](#_Toc140497542)

[References 33](#_Toc140497543)

# List of Figures

[Figure 1: Data fusion framework. (Adetiloye and Awasthi, 2019) 11](#_Toc140507338)

[Figure 2; decision tree implementation in clementine environment. (Shen and Key, 2020) 11](#_Toc140507339)

[Figure 3: Congestion and duration prediction model overview. (Othman, Keoh and Tan, 2017) 12](#_Toc140507340)

[Figure 4: ELM network model. (Method, 2019) 14](#_Toc140507341)

[Figure 5: The LSTM-NN and GRU-NN architectures. (Navarro-Espinoza et al., 2022) 15](#_Toc140507342)

[Figure 6- Methodology 19](#_Toc140507343)

[Figure 7- Boxplot of the vehicles column 27](#_Toc140507344)

[Figure 8- Data after feature engineering 28](#_Toc140507345)

[Figure 9- Histogram distribution for number of vehicles 28](#_Toc140507346)

[Figure 10- Vehicle count wrt hour of the day 29](#_Toc140507347)

[Figure 11- vehicle count by day of week. 29](#_Toc140507348)

[Figure 12- Bar plot of vehicles wrt Junction 30](#_Toc140507349)

[Figure 13-Correlation heatmap 31](#_Toc140507350)

[Figure 14- linear regression forecasting 32](#_Toc140507351)

[Figure 15- Random Forest forecasting 33](#_Toc140507352)

[Figure 16- Gradient boosting forecasting 34](#_Toc140507353)

[Figure 17- Evaluation metrics vs Number of neighbors 34](#_Toc140507354)

[Figure 18- Knn forecasting 35](#_Toc140507355)

[Figure 19- Stacked model forecastings 36](#_Toc140507356)

[Figure 20- Blending model forecasting 37](#_Toc140507357)

# Chapter 1- Introduction

## 1.1 Background

Traffic congestion is one of the major and persistent issues that is faced by most urban areas around the world and with growing cities the population is shifting towards the urban areas which increases the challenges associated with the management of traffic flow and this results in a more complex form of the traffic (Yasir, Nower and Shoyaib, 2022). Congestion on roads not only leads to frustrating delays and travelling for longer hours for several individuals which has a lot of detrimental effects on the environment, economy, and quality of life that people live. Even with this, technological advancements have improved a lot with the increase in resources and a vast amount of data is now available to us which can be used by researchers to study and plan an effective way of solving such problems (Impedovo *et al.*, 2019). The availability of a vast amount of data has helped researchers and transportation experts to be able to develop innovative approaches that help tackle the problems of traffic congestion. The prediction of traffic congestion is one such problem which aims at forecasting and anticipating the areas in which the traffic is generally built up and the areas that get frequently congested.

The improvement of the transportation system is the key that holds the potential to unlock the way or ability with which the prediction of traffic congestion can be performed significantly. Having a traffic congestion prediction system holds a lot of potential for managing traffic and for proactively implementing measures such as rerouting vehicles and optimizing the signal timings which will provide real-time information to the drivers and with that the traffic can be easily mitigated and the congestion can be decreased to enhance the overall traffic efficiency (Razali *et al.*, 2021). There are several factors which come into play when traffic congestion is considered such as the volume of traffic, the conditions of weather, special events and happenings, and accidents. Through the analysis of historical data and with real-time analysis of the data from sensors, traffic cameras, GPS devices and other factors, the predictive models can train efficiently and identify the patterns and trends which enables the anticipation of a congestion-prone area using a reasonable level of accuracy for the model.

However, with the advent of intelligent technologies such as artificial intelligence and machine learning techniques, traffic congestion prediction has been revolutionized. These techniques help us enable the extraction of valuable insights through large datasets and help us to perform very accurate predictions and improved decision-making results (Anjaneyulu and Kubendiran, 2023). These techniques are very efficient in leveraging historical and real-time data which helps machine learning models to learn from the trends and patterns of the data to make predictions which helps in optimizing the traffic management strategies. There are numerous potential benefits of accurate traffic congestion prediction which helps the general public to plan their journey in a safe manner and in a way that they don’t waste their time being stuck in traffic. The commuters or general public can plan their trips for the specific routes in advance which will help them avoid the congested areas and save their valuable time (Mihaita, Li and Rizoiu, 2020). The transportation authorities have a plan for managing the traffic, however, the predictions made by the machine learning models are certainly way more useful. With the implementation of intelligent results, transportation authorities can optimize the deployment of resources which will effectively help in the management of congestion hotspots. These include the deployment of additional police officers in the area who can handle the traffic easily.

A traffic congestion prediction model offers a lot of benefits to the user and the management party to improve transportation management and get enhanced the overall efficiency of the transportation system. some of the key advantages of a traffic congestion prediction system are:

1. It helps in creating efficient travel plans which is one of the major benefits of a congestion prediction system as it allows commuters to plan their journeys effectively. Providing real-time information about congestion-prone areas helps suggest alternative routes and the system enables the drivers to avoid traffic jams and reduce the time of travel (Reshma Ramchandra and Rajabhushanam, 2021).
2. It helps in reducing delays and frustration of people who are stuck in traffic jams as a congestion prediction system will help minimize the delays which will help drivers choose alternative paths and they will be less frustrated (Ban, Guo and Li, 2016).
3. The congestion prediction system provides valuable insights to the transportation authorities which helps in the accurate prediction of high-congestion areas and the deployment of traffic police reduces the congestion in the area (Cheng *et al.*, 2022).
4. It will also help in improving the safety of people travelling on roads as it will warn them ahead of the congestion and people will try to avoid getting stuck which will reduce accidents (Abdullah *et al.*, 2023).
5. It also provides environmental benefits as fewer vehicles will be congested less fuel will be used by the vehicles that are waiting for the traffic to be flowing again and hence less environmental pollution will have resulted (Wang *et al.*, 2018).

In conclusion, it can be said that solving the traffic congestion problem will help a lot of people in their everyday lives and a lot of benefits will be available to the people which will reduce their frustration with people and help their well beings. In this era of digital devices and technologies, smart cities are implementing new techniques which are data-driven and will help in decision-making tasks as a solution to that, the congestion prediction problem has emerged as one of the crucial tools that are required in these times or the quest to provide an efficient and sustainable transportation system to the urban society (Li *et al.*, 2017). The field of traffic prediction continues to evolve at a very fast rate and with the advancements being driven in the field of technology, and with the growth of understanding the complex dynamics of the traffic flow, the smart cities need to grow smarter. Thus, harnessing the power of predictive analysis will provide us with a way to move towards our goal and a future in which traffic congestion can be minimized easily and in which people can effectively enjoy the benefits of a seamless and efficient transportation system (Saleem *et al.*, 2022).

This research study focuses on building a model for traffic congestion prediction, which will use the data from the roads and analyze that data to understand the important features that can be used for predicting the traffic situation. This study uses machine learning techniques for understanding the complex dynamics of the traffic flow and for understanding the research performed in this field, our study has implemented a literature review which provides us with ways to deal with the problem efficiently. The literature review shows us the research gaps which helps us motivate ourselves towards the goals of getting better and better with every researcher working in this field.

## 1.2 Purpose of this study

The purpose of this research is to find and create machine learning algorithms that can accurately forecast when and where traffic congestion will occur. The study's ultimate goal is to develop prediction models that can accurately predict future traffic congestion levels by making use of the dataset containing information on the number of vehicles at various junctions. Using machine learning techniques, this research aims to provide light on how to better manage and distribute traffic, as well as how to better plan for and design transportation infrastructure. The end goal is to increase transportation efficiency and sustainability while decreasing traffic congestion.

## 1.3 Aim and Objectives

The aim of this research is to predict the traffic congestion prediction with the help of machine learning algorithm so that it may improve commute experiences by reducing traffic congestion and improving traffic flow.

The objectives of this research are shown down below.

* Investigating the use of machine learning techniques in Traffic congestion prediction to help with route planning and congestion avoidance.
* To analyze the impact of a holiday or special conditions on traffic congestion and how it affects the accuracy of traffic congestion prediction.
* To evaluate the accuracy of prediction model by comparing predicted congestion levels with actual congestion levels.
* To develop a robust and scalable Traffic congestion prediction model that transportation agencies and city planners can use to improve traffic flow and reduce congestion in the long term.

## 1.4 Outline of the dissertation

The outline of the dissertation given as follows.

* **Chapter 2: Literature Review** - Critically examines prior studies and literature on traffic congestion prediction, identifying gaps and establishing the theoretical framework.
* **Chapter 3: Methodology** - Explains the data collection process, machine learning algorithms utilized, and evaluation metrics employed for traffic congestion prediction.
* **Chapter 4: Implementation Results** - Presents the findings and discusses the performance and outcomes of the implemented models for traffic congestion prediction.
* **Chapter 5: Conclusion and Future Recommendations** - Summarizes the key findings, highlights contributions, and provides recommendations for future research directions in traffic congestion prediction using machine learning algorithms.

# Chapter 2- Literature review

The road network is the backbone of the development structure of any city or a country’s development and therefore free-flowing traffic on roads is important for faster connectivity and transportation. Planning of a smart city is done with the help of maintenance of traffic systems which is an essential way of dealing with and handling the congested areas of the city (Sun, Chen and Sun, 2019). These methods enable us to make extremely precise forecasts, draw important conclusions from big datasets, and offer better decision-making outcomes. These methods may efficiently utilise both historical and current data, allowing machine learning models to learn from the trends and patterns of the data to generate predictions that aid in the optimisation of traffic management strategies (Sunindyo and Satria, 2020). Accurate traffic congestion forecasts might help the public by enabling them to plan safe travel routes and cut down on the amount of time they waste trapped in traffic. Planning travels in advance for certain routes will help commuters and the public avoid crowded regions. This section is efficiently planned, and a broad literature review is presented that is very useful for the research performed in our study.

## 2.1 Machine learning for Traffic congestion prediction

Traffic flow prediction helps in managing the free flow traffic of a city for better road connectivity which helps in faster transportation, and it can be achieved using machine learning algorithms which will be trained for traffic congestion prediction. This research paper (Devi and Neetha, 2017) proposes a set of five machine-learning algorithms for traffic congestion prediction in a smart city based on IoT. Smart cities are equipped with sensors that analyze the flow of traffic and the free flow of road traffic is necessary for well-developed cities. This study uses IoT data and then identifies the path and uses the average speed of vehicles on the same path and compares them against a certain threshold to assign it as congested or non-congested. These two different datasets are grouped, and the machine learning-based algorithm is trained on the above data. The machine learning algorithms which are used for prediction are decision trees, random forests, support vector machine, multilayer perceptron and logistic regression model. (Ma *et al.*, 2015) The data used has five attributes, vehicle id number, timestamp of data, speed of the vehicle, coordinates of vehicles, and congestion in the path. The logistic regression model performs the best among the five models and achieves an accuracy of 99.9%. However, the limitation of this study is that the experiment is not complex and for more adverse and better solutions hybrid models will be implemented.

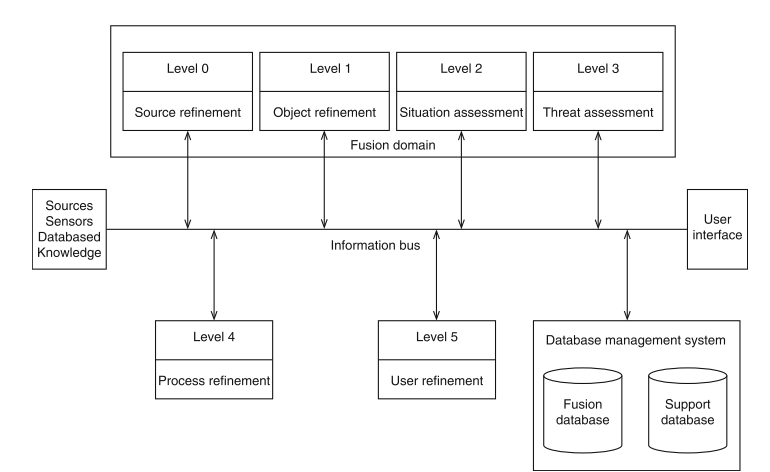


Figure 1: Data fusion framework. (Adetiloye and Awasthi, 2019)

Traffic involves a lot of circumstances and to collect traffic data for traffic congestion prediction, a timely flow of information is required which involves the flow of traffic on the road such as traffic signals, accidents, rallies, and road repairs which might be a cause for traffic. This research study (Meena, 2020) proposes an intelligent way to the prediction of traffic using machine learning models for intelligent transportation systems. to identify the classification and regression, the decision tree model is used where the goal is to predict the value of the target variable. The other detection model that is used is the support vector machine and the random forest model. (Moridpour, 2021) The study uses four features, location, direction, speed and start-end junction which help in identifying the congested situation. After this the congested situation is classified and the random forest model achieves the highest accuracy of 91%. The model performs better as compared to other machine learning models and the limitation of this study is that deep learning and the genetic algorithm haven’t been implemented which could have improved the complexity issues of the model.

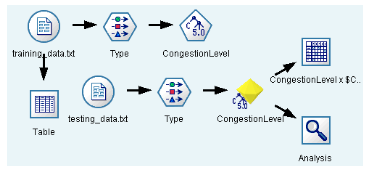


Figure 2; decision tree implementation in clementine environment. (Shen and Key, 2020)

Machine learning models are used for the prediction of traffic congestion and detection in this research study where the approach classifies the traffic into two classes, congested and non-congested or free flow of traffic. The approach (Gatto and Forster, 2021) is based on induction loops, video analysis, horns and GPS or mobile phones using the random forest classifier. The analysis of traffic audio is performed using video recorders where the field of view of the camera is stated at horizontal or vertical angles and the camera is located as per parameters relative to the road. The data is collected in different ways that is through data recordings, downloaded data, sound sources, audio segmentation, derived segments, noise audio, sound analysis and audio feature vectors. The descriptors and feature vectors are used with a block size of 0.5s and a step of 10%. (Elfar, Talebpour and Mahmassani, 2018) The limitation found through the research is that for getting effective results the tracks need to be sufficiently apart from each other on the highway and the sound of vehicles must not interfere with each other. This problem needs to be dealt with as with this problem it would be difficult to predict the situation using two synchronized microphones.

## 2.2 Multimodal framework for the Traffic congestion prediction

Traffic congestion prediction requires the analysis of an enormous amount of data across multiple modalities such as traffic cameras, GPS, and information on the location of vehicles, etc. This research paper (Adetiloye and Awasthi, 2019) proposes a multimodal big data fusion framework that aims at solving the traffic congestion problem based on homogenous and heterogenous data. The fusion model based on homogenous data fuses the data of the same type or in a quantitative manner that is estimated using machine learning models such as random forest, deep belief network and back-propagation neural network model. (Julio, Giesen and Lizana, 2016) The framework also applies the extended Kalman filter that helps with stochastic filtering of non-linear noisiness and therefore reduces the error of measurement and estimation. The other heterogeneous fusion model extends the homogenous model to integrate with the qualitative data such as the traffic tweet information. The results obtained through sentiment analysis and Kalman filter are treated through the Mamdani fuzzy rule inference for the heterogenous traffic data fusion.

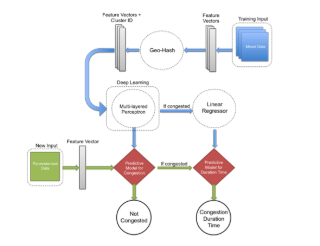


Figure 3: Congestion and duration prediction model overview. (Othman, Keoh and Tan, 2017)

Traffic congestion brings up problems such as delays in work or increased consumption of fuel which leads to additional pollution. This research study (Walraven, Spaan and Bakker, 2016) proposes a traffic flow optimization to deal with congestion management using a reinforcement learning approach. A multimodal big data fusion system that considers both heterogeneous and homogeneous data to solve the problem of traffic congestion. Using machine learning models like the random forest, deep belief network, and back-propagation neural network model, the homogenous data fusion model combines data of comparable sorts or data that has been measured and evaluated. The framework also employs the extended Kalman filter, which assists in the stochastic filtering of non-linear noise and reduces measurement and estimate error. (Liu and Wu, 2017) Incorporating qualitative data, such as input from traffic tweets, the other heterogeneous fusion model expands on the homogeneous model. The results of sentiment analysis and Kalman filtering are handled using Mamdani fuzzy rule inference. Markov decision-making process is used for the description of the state of traffic, the action space, the reward function, the inclusion of traffic predictions and learning policies. (Ata *et al.*, 2020) Apart from the other models, the learning policies such as deep Q networks and neural networks have also been implemented in this study. In-car information displays can first be utilised to offer individualised driving speed recommendations while taking into consideration the expected traffic circumstances. (Kothai *et al.*, 2021) Instead of utilising conventional methods that activate changeable message signs after having identified congestion, this enables proactive traffic flow control. Second, autonomously driving vehicles must choose their ideal speed, and our approach offers a starting point for making such decisions judiciously utilising AI-based algorithms. The conclusion is that this work advances the field of intelligent transportation systems and encourages the use of AI-driven techniques in these practical settings.

## 2.3 Deep learning algorithms for performing the Traffic congestion prediction

While planning a journey a person looks for traffic congestion on the route and then according to that they plan out to travel so that they reach their destination on time and do not have to wait for long hours in the traffic. This study proposes a way for efficient journey planning and performs congestion prediction with the help of deep learning models. The study (Othman, Keoh and Tan, 2017) attempts to delineate the possibilities for the improvement of urban mobility by processing big data and using deep learning models. the paper aims to develop and validate a mobile application to predict traffic conditions that allow road users for making better decisions for their travel plans. This study uses a multi-layered perceptron (MLP) deep learning model for the prediction of congestion and supplements for the linear regression model that helps predict the duration. The MLP-LR model performs the occurrence with an accuracy of 63%. The data uses GPS location for implementation and the model performs training updates through back-propagation to be implemented through the DNN classifier using the TensorFlow model. (Bai *et al.*, 2021) (Kumar *et al.*, 2018) The classifiers used in this study are logistic regression, K-nearest neighbour classifier, stochastic gradient classifier, multi-layered perceptron, naïve Bayes and support vector machine which is tested using the Weka test. The optimizers used for the study are SGD, AdaGrad, AdaDelta and Adam. This study opens several avenues for research and investigations using a real-time application that can prove the possibility of the concept and helps mitigate congestion issues using deep learning.

Planning of routes can be made easy if the road ahead can be promptly predicted and this research study (Li *et al.*, no date) aims to perform congestion prediction using the spatial-temporal correlation and through the evolution characteristics of the traffic flow data using the conv-BiLSTM module that comprises of a convolutional neural network or CNN and through a bidirectional LSTM model. Once the spatial features are extracted from the CNN mode, the temporal features and the alignment features get extracted through the BiLSTM models and then the prediction results are obtained in the form of an output. The BiLSTM model is an RNN model that is used for processing sequential data and compared to ordinary neural networks it helps process sequence-changing data that contains loops for persistent information. (Qi and Cheng, 2023) The model’s performance is compared with other models such as gradient boosting, random forests, CNN, LSTM, Conv LSTM, deep residual networks or ResNet and graph convolutional networks. (Li *et al.*, 2020) The study performs well and achieves better results when compared to existing models. However, there are a few limitations which retards the performance of the study, such as only the basic parameter of the traffic speed is used in this study and integration of more characteristic parameters of traffic flow is required to get better prediction. In addition to that, traffic congestion might be affected by other external or unexpected reasons such as weather, holiday, etc., which might be occasional but needs to be considered for investigation.

Traffic congestion greatly influences the development of urban communities in different ways such as stress levels, fuel wastage, monetary loss and delayed deliveries. This research study (Shen and Key, 2020) performs a comparative study and then builds a machine learning model for the prediction of congestion using the GCM dataset that consists of sixteen urban counties with 2,500 miles of roads. Data processing is performed through TensorFlow and clementine platforms in which the raw sensor data is to be processed in a way for designing an effective and reliable prediction model. The algorithms used for modelling are logistic regression, decision trees and neural networks. The congestion level is classified into four different categories non-congested, light, medium and heavily congested. The predicted value and the actual value are compared against each other for getting verification of the performance of the model. An accuracy of 97% is obtained through the decision tree model which performs best among the other three models. (Martínez-álvarez, 2016) The limitation of this study is that the data doesn’t contain more details about the situation which hinders the application and deep learning models are required to be implemented which would help with the complexity of the dataset.

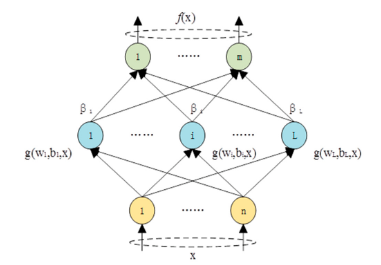


Figure 4: ELM network model. (Method, 2019)

Prediction of large-scale traffic congestion is tough to achieve and for that purpose, this research study (Method, 2019) applies a symmetric extreme learning machine cluster for a fast learning method so that traffic congestion is predicted accurately and people travelling by road can plan their journey with ease. Neural networks have proved their worth when implemented through machine learning models and this study uses models such as extreme learning machines, symmetric extreme learning machines and cluster fast learning methods for training the models. the data is processed and that feature extraction is performed with the help of the section clustering method that checks for the conditions of roads the feature extraction uses the road factors, environmental factors and sudden factors for the prediction. (Park *et al.*, 2009) The cluster model is put through a tuning test using three parameters which are, length of time series, regularization coefficient, number of hidden layer nodes and comparison test for symmetric and non-symmetric ELM models against other models such as GBDT, ridge, lasso, logistic regression and support vector machine. The algorithm achieved an accuracy of 93% which is higher than other models but it needs to be tested on real-time data and used for real-time purposes and validate to check if it is working correctly and efficiently.

Revealing road conditions along the route also helps to prevent accidents and road congestion and since roads nowadays are equipped with a huge amount of surveillance cameras, it can be used for real-time identification of vehicles which provides traffic flow estimation. This research (Navarro-Espinoza *et al.*, 2022) provides a way to control traffic using remote-controlled traffic lights and for that purpose machine learning and deep learning methods are implemented for solving this issue. In this study, two public datasets are used for training the models where the first one contains the number of sampled vehicles every five minutes and at six intersections for 56 days. Two recurrent neural network models have been implemented in this study, the GRU and LSTM model where mean squared error is used as a loss function and RMSprop is used as an optimizer. (Khan *et al.*, 2019) For classification, the machine learning model are used which are, linear regression, gradient boosting classifier, stochastic gradient descendent regressor, multilayer perceptron regressor and the random forest regressor model. For evaluation, metrics such as MAE, MAPE, RMSE, explained variance and r squared are used. The RNNs and random forest regression model have similar scores and the results were satisfactory to predict traffic flow on the four lanes of intersection which demonstrates the feasibility of it being implemented on smart traffic light controllers.

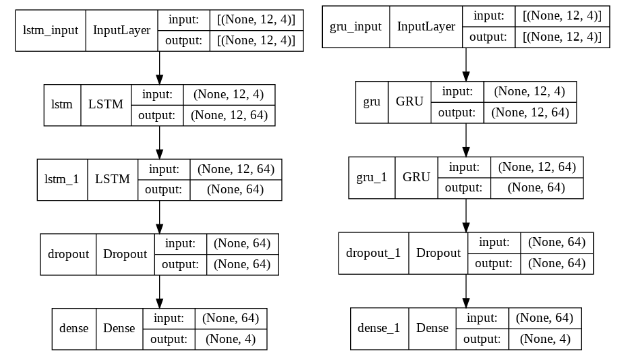


Figure 5: The LSTM-NN and GRU-NN architectures. (Navarro-Espinoza et al., 2022)

## 2.5 Research gaps

In this section literature review has been performed where a wide group of papers has been reviewed and with this analysis, a lot of conclusions about the research going on in the field of traffic congestion management can be deduced. The common research gap faced by the machine learning model is that the experiment is not complex and for more adverse and better solutions hybrid models will be needed and implemented. Another important piece of consideration for improving the system predicts that only the basic parameter of the traffic speed is used in this study and integration of more characteristic parameters of traffic flow is required to get better prediction. In addition to that, traffic congestion might be affected by other external or unexpected reasons such as weather, holiday, etc., which might be occasional but needs to be considered for investigation. Machine learning algorithms are incapable of keeping up with the complexity required to be solved, but deep learning and the genetic algorithm haven’t been implemented that could have improved the complexity issues of the model. Another leading research gap is that the data doesn’t contain more details about the situation which hinders the application and deep learning models are required to be implemented which would help with the complexity of the dataset. These research gaps are very important to consider for improving the results of an experiment and need to be analysed in such a way that the experiment is capable of getting efficient results.

Table 1- Various comparison of research paper

|  |  |  |  |
| --- | --- | --- | --- |
| Reference | Approach | Results | Limitations |
| (Devi and Neetha, 2017) | A set of five machine learning algorithms for traffic congestion prediction in a smart city based on IoT. decision trees, random forest, support vector machine, multilayer perceptron and logistic regression model | The logistic regression model performs the best among the five models and achieves an accuracy of 99.9%. | the limitation of this study is that the experiment is not complex and for more adverse and better solutions hybrid models will be implemented. |
| (Adetiloye and Awasthi, 2019) | random forest, deep belief network and back-propagation neural network model are used | The results obtained through sentiment analysis and Kalman filter are treated through the Mamdani fuzzy rule inference for the heterogenous traffic data fusion which improves the prediction. | The lack of an appropriate dataset is seen which is a main concern and the complexity of the model can be solved. |
| (Othman, Keoh and Tan, 2017) | multi-layered perceptron and logistic regression are implemented along with several machine learning classifiers. | This study opens several avenues for research and investigations using a real-time application that is capable of proving the possibility of the concept and helps mitigate congestion issues using deep learning. | No limitations of this study have been registered. |
| (Li *et al.*, 2019) | Conv-BiLSTM model with CNN and bidirectional LSTM model is implemented. | The study performs well and achieves better results when compared to existing models. | Only the basic parameter of the traffic speed is used in this study and integration of more characteristic parameters of traffic flow is required to get better prediction. In addition to that, traffic congestion might be affected by other external or unexpected reasons such as weather, holiday, etc., which might be occasional but needs to be considered for investigation. |
| (Meena, 2020) | A decision tree, support vector machine and random forest model are implemented. | The model performs better as compared to other machine learning models while achieving an accuracy of 91%. | Deep learning and the genetic algorithm haven’t been implemented which could have improved the complexity issues of the model. |
| (Shen and Key, 2020) | Algorithms used for modelling are logistic regression, decision trees and neural networks | An accuracy of 97% is obtained through the decision tree model which performs best among the other three models. | the data doesn’t contain more details about the situation which hinders the application and deep learning models are required to be implemented which would help with the complexity of the dataset. |
| (Method, 2019) | Extreme learning machines, symmetric extreme learning machines and cluster-fast learning methods are implemented. | The algorithm achieved an accuracy of 93% which is higher than other models. | The limitation of this study is that more algorithms should have been compared and the results could have been enhanced through the use of advanced techniques. |
| (Navarro-Espinoza *et al.*, 2022) | Two recurrent neural network models have been implemented in this study, the GRU and LSTM model along with other machine learning regressors. | The RNNs and random forest regression model have similar scores and the results were satisfactory to predict traffic flow on the four lanes | The limitation of this study is that the model is incapable to predict in cases where the traffic is very high and the remote traffic light cannot be operated. |
| (Gatto and Forster, 2021) | A random forest classifier is implemented. | This approach is based on induction loops, video analysis, horns and GPS or mobile phones where the audio analysis of vehicles is performed. | The limitation of this study is that for getting effective results the tracks need to be sufficiently apart from each other on the highway and the sound of vehicles must not interfere with each other. |
| (Walraven, Spaan and Bakker, 2016) | Markov decision-making process is used for deep Q network and neural networks. | In-car information displays can first be utilised to offer individualised driving speed recommendations while taking into consideration the expected traffic circumstances. | No limitation has been mentioned for this research study. |

## 2.6 Summary

In summary, the literature review section of this paper provides a thorough evaluation of previous research on the topic of traffic congestion prediction using machine learning techniques. It takes a close look at the theories, methods, and models now in use for predicting traffic congestion. This chapter provide a critical analysis of previous work and to provide the foundations for the technique and execution, as well as to identify any gaps or shortcomings in the available literature.

# Chapter 3- Methodology

In this chapter the methodology of the project will be given in details. It covers from data collection to model evaluation and tunning.

A diagram of methodology

Description automatically generated

Figure - Methodology

## 3.1 Data collection

The dataset used for this dissertation was obtained from Kaggle (fedesoriano, no date), a popular platform for sharing and discovering datasets. It consists of 48.1k (48,120) observations of the number of vehicles recorded each hour in four different junctions. The dataset includes the following variables:

* DateTime: Represents the date and time of the observation.
* Junction: Indicates the junction number where the data was collected.
* Vehicles: Represents the count of vehicles observed at each hour.
* ID: Unique identifier for each observation.

The data collected comes from different time period, as the sensors on each junction collected data at different times. Some junctions may have limited or sparse data, requiring careful consideration when analyzing the dataset and making future projections.

## 3.2 Ethical Considerations

Ethical considerations play a crucial role in any research project. In the context of this dissertation on traffic congestion prediction using machine learning algorithms (de Harder, 2023), there are several ethical considerations to address:

* **Privacy and data protection**: Ensure compliance with regulations and secure handling of data.
* **Transparency and informed consent:** Obtain consent and ensure transparency in data usage.
* **Fairness and bias**: Address biases in data and models to avoid discrimination.
* **Transparency and explainability**: Make models transparent and explainable to stakeholders.
* **Continual monitoring and improvement**: Regularly assess ethical concerns and refine models accordingly.
* **Promote ethical practices**: Incorporate these considerations to uphold ethical standards in traffic congestion prediction using machine learning.

This dissertation has the potential to further the development of responsible and ethical practices in traffic congestion prediction using machine learning algorithms by including these ethical issues into the research process.

## 3.3 Requirement Specifications

The following requirements must be met in order for the machine learning algorithms traffic congestion prediction project to work smoothly:

1. First, set up Jupyter Notebook, a shared, collaborative workspace for writing code and experimenting with its visualisations.
2. Numpy, Pandas, Matplotlib, Seaborn, and Scikit-Learn are just some of the Python libraries you'll need to get started with this project. Package managers like pip and conda can be used to set up these libraries.
3. Minimum 4 GB of RAM is required to process the data and train the models efficiently.

By meeting these requirement specifications, you will have the necessary environment and tools to execute the traffic congestion prediction project in Jupyter Notebook with appropriate computational resources and essential Python libraries.

## 3.4 Checking the quality of Data

Data quality analysis is a vital part of any data analytic endeavour. Data quality assurance involves verifying that the data is comprehensive, accurate, consistent, and has good integrity before using it for analysis. By evaluating data quality, potential issues and errors can be identified and addressed, ensuring the validity and trustworthiness of the analysis, and resulting insights (Samuel, 2022).

There are following steps which are taken to check the quality of data.

* The first step is to check for data entry error such as incorrect values, or typos error in the dataset. These errors can occur during data collection or recording and can affect the accuracy and reliability of the data.
* The second step is to check for the missing values and duplicate values. Missing values can introduce bias or impact analysis results, while duplicate values can distort statistical summaries and lead to inaccurate conclusions, making it crucial to identify and handle them appropriately.
* The third step is to check for the outliers present in the dataset. Identifying and dealing outliers helps maintain the integrity of the data analysis by reducing the impact of extreme values that might skew results or lead to erroneous interpretations.

## 3.5 Data Preprocessing

Cleaning and preparing data for analysis are foundational activities for every machine learning or data science activity. These procedures involve cleaning and structuring data so that it can be used in further modelling and analysis (Awan-Ur-Rahman, 2019). There are the following steps which are followed to clean the dataset.

* If the data contains data entry error, it can be either removed or replaced with the correct values.
* If there are missing values present in the dataset then perform imputation techniques such as mean, mode or machine learning algorithm like KNN. If the dataset contains duplicate values, then you must drop the duplicate values from the dataset.
* Outlier removal is not important for the traffic congestion prediction because there may be on some days it represent abnormal or extreme traffic conditions that deviate from the regular patterns, removing them may result in a loss of important information.
* Perform feature engineering on the columns present in the dataset. Create new features or transform existing features to enhance the predictive power of the models. This can involve deriving additional variables from existing ones, such as extracting day of the week or hour of the day from DateTime or creating interaction terms between variables.
* After the dataset is processed it will be splitted into training and testing. This split ensures that the model is assessed on its ability to generalize to new instances and helps to estimate its real-world performance.

## 3.6 Machine learning algorithms

In this section all the machine learning algorithms will be discussed thoroughly for the traffic congestion prediction.

### 3.6.1 Linear regression

Linear regression is a statistical algorithm used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the variables, aiming to find the best-fitting linear equation that represents the data (Swaminathan, 2018). The equation takes the form of:

Y = b0 + b1\*X1 + b2\*X2 + ... + bn\*Xn

where Y is the dependent variable, b0 is the intercept, b1 to bn are the coefficients (also known as weights), and X1 to Xn are the independent variables.

Estimated coefficients are those that lead to the smallest sum of squared deviations between the predicted and actual values in the training data. Ordinary Least Squares (OLS) is a technique that does this by finding the coefficients that produce the least residuals.

The coefficients are estimated using the following formulas:

b1 = Σ((Xi - X̄)(Yi - Ȳ)) / Σ((Xi - X̄)²)

b0 = Ȳ - b1\*X̄

where Xi and Yi are the values of the independent and dependent variables, respectively, X̄ is the mean of the independent variable, and Ȳ is the mean of the dependent variable.

Linear regression is an important technique for this study due to its accessibility and simplicity of interpretation. This allows us to analyse the correlation between the number of vehicles and the independent variables of time and junction. Using the linear equation, one may make predictions regarding traffic congestion and evaluate the effect of each independent variable by forecasting the coefficients.

### 3.6.2 Random Forest regression

The ensemble learning approach known as Random Forest Regression uses a forest of decision trees to draw conclusions. This algorithm is a variant of the random forest technique, modified specifically for use in regression analysis.

During training, each tree learns to predict the target variable based on different subsets of the data and features. The final prediction is obtained by averaging or taking the majority vote of the predictions from all the individual trees (Raj, 2020).

Mathematically, let's consider a dataset with N observations and M features. In the random forest regression algorithm:

1. Random subsets of the dataset (bootstrapped samples) are created by randomly sampling N observations with replacement.
2. For each subset, a decision tree is built by recursively splitting the data based on the selected features. The splits are determined by minimizing the variance or mean squared error in the target variable.
3. The trees are grown deep without pruning to capture complex relationships in the data.
4. A prediction is made by adding up the values from each tree in the forest. Predictions are often averaged for use in regression tasks.

Overfitting is mitigated and generalization performance is enhanced by the algorithm's use of randomization in feature and data subset selection. It improves the model's prediction performance by letting it account for additional variables and variances in the data.

The ability of Random Forest Regression to deal with non-linear correlations, high-dimensional data, and interactions between characteristics makes it particularly useful in this context. It can make accurate forecasts based on traffic congestion data by integrating numerous decision trees. Random forest regression can also effectively deal with noisy or missing data, and it has a low overfitting risk to boot. Since a wide variety of factors can affect traffic patterns, it is well-suited for jobs involving congestion prediction. It can process both numerical and categorical features.

### 3.6.3 Gradient boosting regressor

The ensemble learning approach known as Gradient Boosting Regressor sequentially integrates numerous weak prediction models, most often decision trees. It enhances prediction accuracy by gradually training an ensemble of models to correct the mistakes of one another (Masui, 2022).

The algorithm works as follows:

1. Initialize the model with a simple model, such as a decision tree or constant value, which serves as the initial prediction for all instances in the training data.
2. Compute the residuals (difference between the actual values and the predicted values) for each instance in the training data based on the current model.
3. Train a weak learner, usually a decision tree, on the residuals. The weak learner is fitted to the negative gradient of the loss function with respect to the current model's prediction.
4. Update the model by adding the predictions of the weak learner to the current model's predictions. This process is done by multiplying the predictions with a learning rate to control the contribution of each weak learner.
5. Repeat steps 2 to 4 for a specified number of iterations (number of weak learners), adjusting the residuals at each step to minimize the overall loss.
6. The final prediction is obtained by summing the predictions of all weak learners.

Gradient Boosting Regressor is important in this project as it can handle complex patterns and interactions in the data. Over time, the model is refined based on the gradients of the loss function, making it more accurate and predictive by fixing its flaws. Among the many variables and circumstances that contribute to traffic congestion, this algorithm excels in capturing them all and making accurate predictions.

### 3.6.4 K nearest neighbour

As a non-parametric approach, K-Nearest Neighbours (KNN) can be applied to both classification and regression problems. For the purposes of regression, KNN uses the values of the k nearest neighbours to make a prediction about the value of a data point. It uses the majority vote or the average of the neighbouring points to determine a classification or regression, respectively (Chelliah, 2020).

These are the steps of the algorithm:

* Step 1: The first step is to find the optimal value for k, the number of nearest neighbours.
* Step 2: Determine how far off the data point being predicted from is from every other data point in the training set. Euclidean distance and the Manhattan distance are two common measures of separation.
* Step 3: Third, using the determined distances, choose k nearest neighbours.
* Step 4: When performing regression, take the mean of the value of the dependent variable among the k nearest neighbours. For the specified data point, this mean is used as a prediction.

K-Nearest Neighbours is useful for this task because it can detect contextual dependencies and trends in the data at the neighbourhood level. It extrapolates future values from the training data based on the averages of comparable events. When making predictions about traffic congestion, KNN can take into account nearby instances that share comparable traffic conditions. This technique shines when there are temporal or spatial patterns in the data, and it can shed light on regional patterns of traffic congestion from close by cases.

### 3.6.5 Stacking model

To improve prediction accuracy, stacking utilizes numerous base models (Brownlee, 2020) in an ensemble learning setting. In the stacking model, the dataset is used to train multiple independent base models, such as a RandomForestRegressor and a GradientBoostingRegressor. Their forecasts are then fed into a meta-model, often a linear regression model, as independent features.

The algorithm works as follows:

1. Create a validation set and a training set from the dataset.
2. Use the training data to fine-tune your random forest and gradient boosting baseline models.
3. The validation set is used to generate predictions from each base model.
4. Add the original features to the forecasts from the base models.
5. Using the target variable from the validation set, train the meta-model (linear regression) on the combined predictions and the original features.
6. Predictions can be made by first running the test data through each basic model. Then, feed the combined predictions and raw features into the trained meta-model to generate a final prediction.

Because of its ability to incorporate the predictions of various base models, the stacking model is crucial to this endeavour. The stacking model seeks to capture supplementary information from many base models and produce a more robust and accurate prediction for traffic congestion by training a meta-model on the combined predictions and original features.

### 3.6.7 Blending model

Blending, like stacking, is an ensemble learning method that combines numerous base models to create predictions. RandomForestRegressor and GradientBoostingRegressor are two examples of the base models that are trained independently of one another in the blending model. A meta-model (linear regression) is trained on the blended predictions obtained using a weighted average approach to combining the predictions of the base models (Brownlee, 2020).

The algorithm works as follows:

1. Create a validation set and a training set from the dataset.
2. Use the training data to fine-tune your random forest and gradient boosting baseline models.
3. The validation set is used to generate predictions from each base model.
4. Use a weighted average method to combine the predictions from the base models, with the weights representing the relative importance of the forecasts from each base model.
5. Use the validation set's target variable to train the meta-model (a linear regression) on the blended predictions.

The blending model plays a crucial role in this endeavour since it permits the weighted averaging of predictions from multiple base models. The goal of the blending model is to produce a more reliable and accurate traffic congestion prediction by training a meta-model on the blended predictions, thus capturing the strengths and diversity of the base models. The blending model can optimise the final prediction performance by weighting the contribution of each base model appropriately using the weighted average approach.

## 3.7 Hyperparameter tunning

The performance of machine learning models can be greatly enhanced by tweaking their hyperparameters. In the Python ecosystem, GridSearchCV is a popular tool and technique for fine-tuning hyperparameters. In order to identify the best set of hyperparameters, GridSearchCV iteratively evaluates the performance of the model with cross-validation and picks the combination that delivers the best results. GridSearchCV requires a set of hyperparameters and a range of values to be defined before it can be used. These hyperparameters can be used to modify a wide range of model parameters, such as the learning rate, regularisation strength, number of layers, kernel size, etc. For each feasible hyperparameter setting, the model is trained and evaluated via GridSearchCV (Lee, 2021).

The second argument, n\_jobs, determines how many jobs will run in parallel during the grid search. When working with a sizable dataset or intricate models, this parameter's ability to leverage several CPU cores can significantly speed up the process of exploring the hyperparameter space.

GridSearchCV also includes cross-validation as a crucial parameter. With cross-validation, the dataset is split into smaller subsets, and the model is trained and assessed on a variety of these smaller subsets, or "folds." By reducing the possibility of overfitting or underfitting, cross-validation provides a more accurate evaluation of the model's efficacy. Finally, the scoring metric that will be used to rank the effectiveness of the model is defined. Common measures for scoring include the square root of the mistake. GridSearchCV finds optimal hyperparameters according to the intended model performance using the scoring metric.

Hyperparameter space searches like GridSearchCV, which evaluates every possible combination, can be quite computationally intensive. However, it guarantees a comprehensive investigation and the choice of the best hyperparameters according to the given scoring criteria.

## 3.8 Evaluation metrics

There are 3 evaluation metrics which are used to evaluate the traffic congestion prediction model (Raj, no date).

* **MSE (Mean Squared Error)** measures the average squared difference between the predicted values and the actual values in a regression task. It is calculated as:

MSE = (1/n) \* Σ(yi - ŷi)^2

* **RMSE (Root Mean Squared Error)** is the square root of the MSE and provides a more interpretable metric. It is calculated as:

RMSE = √(MSE)

* **MAE (Mean Absolute Error)** measures the average absolute difference between the predicted values and the actual values in a regression task. It is calculated as:

MAE = (1/n) \* Σ|yi - ŷi|

## 3.9 Summary

Several phases of the project are covered in the methodology section. Ethical and requirement considerations were made during data collecting. After ensuring the data was of sufficient quality, it was preprocessed. Linear regression, random forest regression, gradient boosting regressor, K closest neighbour, stacking model, and blending model were only some of the machine learning methods used. The results of the test were used to determine the best course of action. Finally, the models' efficacy was measured using evaluation measures.

# Chapter 4- Implementation, Results and Discussion

## 4.1 Implementation and Results

First, the dataset is loaded and load with the help of pandas library. With the help of shape function, you can find shape which is 48120 rows and 4 columns. There are three integer columns are there and one object column is present in the dataset.

The Traffic congestion dataset does not contain any missing or duplicate values in it. However, there are few outliers present in the Outlier column. The outliers will not be dropped because it will lead to the loss of important information from the dataset.

A graph of a box plot

Description automatically generated

Figure 7- Boxplot of the vehicles column

Next, the DateTime column is converted into the pandas DateTime format. With the help of features engineering techniques there are many columns which are extracted from the DateTime column such as Year, Month, Day, Day of Week, hour and minute.

Next the data is aggregated on an hourly basis using the groupby() function. It allows us to capture more detailed temporal patterns and variations in the number of vehicles. It is also quite useful when dealing with time series data that exhibit hourly fluctuations or when there is a need to analyze data at a more granular level than daily or monthly.

After that two new lagged features are created and added to the original dataframe. These columns capture the lagged values of the 'Vehicles' column by shifting the values by 1 and 3 time steps, respectively. These features can be very useful for capturing temporal dependencies and trends in the data.

Next, the rolling mean and standard deviation are calculated using the 'Vehicles' column over a window of 3-time steps. These columns can help identify trends, seasonality, and variations in the data over time. After that the columns which are not helpful such as DateTime or ID dropped from the dataset.

A screenshot of a computer

Description automatically generated

Figure 8- Data after feature engineering

These missing values will later be filled with zeros before doing machine learning modelling.

Once the data preprocessing steps are completed the next step is to perform the Exploratory data analysis on the Traffic congestion dataset to find the relevant insights from the dataset.

First, the distribution of vehicles is plotted with the help of Histogram. It shows that the plot is having right skewed distribution it means that most of the values are on the left side only.

A graph of distribution of vehicles

Description automatically generated

Figure 9- Histogram distribution for number of vehicles

Next, a bar plot is generated to visualize the relationship between the 'Vehicles' variable and the 'Hour' variable. The plot presents the average number of vehicles at different hours of the day. By examining the bar plot, you can gain insights into the fluctuations and patterns in traffic volume throughout the day, enabling the identification of peak hours and any noteworthy trends based on the hour of the day. The bar plot is shown down below.

A graph of different colored bars

Description automatically generated

Figure 10- Vehicle count wrt hour of the day

The bar graph that was made shows that the evening, between 5 and 11 o'clock, is when most vehicles are on the road. During this time, there are a lot more vehicles on the road than at other times of the day. On the other hand, there aren't as many vehicles as possible on the road in the morning. Based on this data, there seems to be a trend of more traffic congestion and activity in the evening, which could be caused by things like rush hour or people coming home from work. The smaller number of vehicles in the morning may mean that there is less traffic or less going on at that time. These observations give important information about how vehicles are spread out in time throughout the day. This helps with traffic planning, allocating resources, and knowing traffic patterns.

Next, a bar plot is made to examine the relationship between the 'Vehicles' variable and the 'DayOfWeek' variable. The plot represents the average number of vehicles for each day of the week. The bars are ordered in descending order based on the mean 'Vehicles' values, allowing us to identify any variations in traffic volume across different days of the week.

A graph of different colored bars

Description automatically generated

Figure 11- vehicle count by day of week.

There are the following insights obtained from the above bar plot.

* The highest average number of vehicles is observed on Mondays (represented by 1) and Wednesdays (represented by 3), suggesting increased traffic volume during the middle of the week.
* Tuesdays (represented by 2) also show a relatively high average vehicle count, indicating sustained traffic levels.
* Thursdays (represented by 0) exhibit a lower average number of vehicles, potentially indicating a slight decline in traffic volume.
* Fridays, Saturdays, and Sundays (represented by 4, 5, and 6, respectively) demonstrate a gradual increase in average vehicle counts, indicating higher traffic volumes during the weekends.

These findings offer insights into weekly traffic patterns, potentially aiding in transportation planning, resource allocation, and understanding traffic behavior based on different days of the week.

The bar plot reveals the relationship between the 'Vehicles' variable and the 'Junction' variable.

A graph of a number of vehicles by junction

Description automatically generated

Figure 12- Bar plot of vehicles wrt Junction

Based on the plot, it can observe that junction 1 has the highest average number of vehicles, indicating it experiences the highest traffic volume among the four junctions. Conversely, junction 4 has the lowest average number of vehicles, suggesting it experiences the least traffic congestion compared to the other junctions.

At last, the correlation table is made to presents the relationships between the variables in the dataset. The analysis reveals that there is a weak positive correlation (0.22) between the number of vehicles and the hour of the day, indicating that vehicle counts may slightly increase as the day progresses. Additionally, there is a slight negative correlation (-0.13) between the number of vehicles and the day of the week, implying a minor decrease in vehicle counts as the week progresses. The most significant correlation is observed between the number of vehicles and the junction (-0.61), indicating a strong negative relationship. This suggests that certain junctions experience higher traffic congestion compared to others, resulting in lower vehicle counts.

A diagram of heatmap

Description automatically generated

Figure 13-Correlation heatmap

Once the data preprocessing and Exploratory Data Analysis is completed the next step is to split the dataset into training and testing for the purpose of machine learning modelling.

**Machine learning modelling**

In this six-machine learning model are trained and tuned in order to predict the Traffic congestion. After that model will be evaluated using regression metrics like mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE).

**Linear regression**

First, an instance of the LinearRegression model is created. After that model is trained using the training data and make the prediction on the test data. After that model performance was evaluated on the test dataset using metrics like MSE, RMSE and MAE.

Hyperparameter tuning is typically not performed for linear regression because linear regression models have limited hyperparameters to tune compared to other models like random forests or gradient boosting.

A graph of a graph

Description automatically generated

Figure 14- linear regression forecasting

**Random forest regressor**

Similarly Random Forest instance is created. After that model is trained and evaluated on the test data. Then the performance of the model was evaluated using regression metrics.

After that Hyperparameter tunning is performed with the help of Grid search CV library to find the optimal parameters of the Random Forest model. Three parameters are tuned which are number of estimators, max depth and min samples split. Next the model was tuned with 5-fold cross validation and scoring metrics is the negative mean squared error and the value of n\_jobs is set to -1 it means all the CPU cores are used for tunning purpose.

The best hyperparameters found are {'max\_depth': None, 'min\_samples\_split': 2, 'n\_estimators': 300}.

After that tuned model is used to make the prediction on the test set. The performance metrics were relatively lower as compared to the without tuned model. It means by the help of hyperparameter tunning the random forest model performance got improved.

A graph of a graph

Description automatically generated

Figure 15- Random Forest forecasting

**Gradient boosting regressor**

Next Gradient boosting instance was created. After that model trained on training data and evaluated on the test data. Next the performance of model was evaluated and hyperparameter tunning was performed to find the optimal sets of parameters which leads to the best performance. Three parameters are selected for tunning purpose which are number of estimators, learning rate and mas depth. After tunning the best hyperparameter found are {'learning\_rate': 0.1, 'max\_depth': 5, 'n\_estimators': 300}

The prediction on the test set is then made using the tweaked model. After compared to the untuned model, the performance metrics were lower after tuning was applied. This signifies that the performance of the gradient boosting regressor model was enhanced using hyperparameter tuning.

A graph of a graph

Description automatically generated

Figure 16- Gradient boosting forecasting

**K nearest neighbors**

First, the Elbow plot was used to find the best values of K which represents the number of neighbors. It iterates over different values of k to evaluate the KNN model's performance.

A graph of a number of neighbors

Description automatically generated

Figure 17- Evaluation metrics vs Number of neighbors

The plots show at k=5 the model is giving the best performance on the test data. Next the hyperparameter tunning is performed to improve the performance of KNN model even further. The weights parameter is taken for tunning. The best hyperparameter found are {'weights': 'distance'}.

After that tuned model is made prediction on test set and evaluates using various regression metrics. The results shows that model performance got improved after tunning.

A graph of a number of vehicles

Description automatically generated

Figure 18- Knn forecasting

**Stacked model**

A stacking ensemble is created using the base models (Random Forest, Gradient Boosting) and a meta model (Linear Regression). The base models are trained using the training data and predictions from the base models are combined and used as inputs for the meta model. The stacking model is trained using the combined predictions and the target variable and predictions are made using the stacking model on the test data. After that model is evaluated with the regression metrics like RMSE, MSE and MAE.

A graph of a graph

Description automatically generated

Figure 19- Stacked model forecastings

**Blending model**

Similar to the stacking model, a blending ensemble is created using the base models. The base models (Random Forest, Gradient Boosting) are trained individually using the training data. Predictions from the base models on the training data are collected and the predictions from the base models are combined as features. The meta model (Linear Regression) is trained using the combined predictions and the target variable predictions are made using the blending model on the test data. After that model is evaluated with the regression metrics like RMSE, MSE and MAE. `

A graph of traffic congestion forecasting

Description automatically generated

Figure 20- Blending model forecasting

Table 2- comparison of the model performance

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | MSE | RMSE | MAE |
| Linear regression | 7.28 | 2.69 | 1.69 |
| Random forest regressor | 4.13 | 2.03 | 1.15 |
| Gradient boosting regressor | 3.66 | 1.91 | 1.11 |
| K nearest neighbors | 7.43 | 2.72 | 1.82 |
| Stacked model | 4.22 | 2.05 | 1.22 |
| Blending model | 4.32 | 2.08 | 1.17 |

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) values can be used to compare the performance of various machine learning models. Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, K Nearest Neighbours, Stacked Model, and Blending Model are the six models available here. Based on these metrics, let's evaluate their performance.

To begin, we look at the MSE, which is the average squared difference between the anticipated and actual values and see that the Gradient Boosting Regressor has the best overall performance by minimising prediction errors with an MSE of 3.66. The next two closest methods, the Random Forest Regressor, and the Stacked Model, both have MSE values of 4.13. Linear Regression and the Blending Model both have MSE values of 7.28, while K Nearest Neighbours has an MSE of 7.43, making it the most inaccurate method. Therefore, when comparing models using MSE, Gradient Boosting Regressor is the winner.

The Gradient Boosting Regressor also has the lowest root-mean-squared-error (RMSE) value (1.91), which is a measure of the average prediction error (MSE). This indicates that it makes the most precise predictions, with RMSE values of 2.03 and 2.05, respectively, followed by the Stacked Model and the Random Forest Regressor. When compared to Linear Regression and the Blending Model, K Nearest Neighbours has the greatest RMSE value (2.72), with RMSE values of 2.08 and 2.69 respectively. Therefore, the Gradient Boosting Regressor continues to excel in terms of root-mean-squared-error (RMSE).

Last but not least, when looking at the MAE, which is the average absolute difference between the anticipated and actual values, we find that the Gradient Boosting Regressor has the shortest MAE of 1.11. Following closely behind with MAE values of 1.15 and 1.17, respectively, are the Random Forest Regressor and the Blending Model. While Linear Regression and K Nearest Neighbours both have MAE values of 1.69, the Stacked Model's MAE is only 1.22. The Gradient Boosting Regressor once again displays excellent MAE performance.

Overall, based on the given performance metrics, the Gradient Boosting Regressor consistently outperforms the other models in terms of MSE, RMSE, and MAE. It achieves the lowest error values, indicating superior accuracy and predictive power. The Random Forest Regressor, Stacked Model, and Blending Model also perform relatively well, while Linear Regression and K Nearest Neighbors exhibit higher prediction errors while making traffic congestion prediction.

## 4.2 Discussion

In this section the detailed discussion of the whole experiment will be given. It involves the strength and some certain limitation related to this experiment.

The first step in ensuring the quality and reliability of the dataset is the extensive data preprocessing process. Relevant information and trends in the data are captured by the model thanks to the incorporation of feature engineering techniques such as the elimination of missing or duplicate values and the extraction of additional temporal features and the calculation of lagged features. In addition, reducing the number of columns in the dataset helps to eliminate extraneous data and facilitates more precise analysis.

Second, the methodology's use of exploratory data analysis (EDA) yields interesting findings from the data collection. The EDA reveals traffic congestion patterns and their correlations between elements including time of day, day of week, and intersection through the use of visualisations like histograms and bar charts. The model's findings are useful for managing traffic congestion since they help with things like traffic planning, resource allocation, and analysing driver behaviour.

The ability to compare and contrast several ML models is another strength of the approach. The methodology provides for a comparison of various approaches by evaluating models such as Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, K Nearest Neighbours, Stacked Model, and Blending Model. The model's performance may be statistically evaluated with regression metrics like MSE, RMSE, and MAE, allowing for the most accurate and applicable model to be selected.

Among the investigated models, the Gradient Boosting Regressor stands out for its superior performance across all regression criteria. This demonstrates that the model is quite accurate in predicting traffic delays. According to the results of this research, the Gradient Boosting Regressor is the best model for foreseeing traffic congestion in this particular dataset. Its strength is in its ability to reliably predict traffic congestion thanks to its high level of accuracy and performance. Hyper-parameter tuning with GridSearchCV is also a part of the process, and it's used on models like the Random Forest Regressor and the Gradient Boosting Regressor. This method aids in locating appropriate hyperparameter values, which in turn improves the performance of the models. The process enhances the accuracy and predictive capacities of the chosen models by means of fine-tuning.

However, there are few limitations found while doing this project. First, there is a lack of features specifically relevant to traffic congestion prediction in the dataset. Traffic congestion can be caused by a variety of reasons, some of which may not be accounted for in the dataset utilised. These factors include, for example, weather, road conditions, and real-time traffic data. The model's ability to anticipate traffic congestion in real-world circumstances may be hindered without these key components. The performance and predictive abilities of the model could be enhanced by investigating alternative data sources or collecting more extensive datasets that cover a greater variety of attributes.

Additionally, the methodology primarily focuses on a limited set of machine learning models, namely Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, K Nearest Neighbors, Stacked Model, and Blending Model. While these models cover a range of approaches, there may be other models or ensemble techniques that could potentially provide better predictions for traffic congestion. Exploring alternative models, such as neural networks, support vector regression, or time series models, could provide a more comprehensive analysis and potentially improve the accuracy of the predictions.

In summary, the experiment proves that machine learning models are capable of accurately forecasting when and where traffic congestion would occur. Large amounts of time were spent on data preprocessing, exploratory data analysis, and model evaluation, all of which improved confidence in the results. As the best-performing model, the Gradient Boosting Regressor demonstrates its prowess in making reliable traffic congestion forecasts. However, there are restrictions, such as a restricted choice of machine learning models and the absence of traffic congestion-specific variables in the dataset. Improving the prediction abilities requires addressing these restrictions through the addition of more relevant features and the exploration of alternative models. In conclusion, this study demonstrates how machine learning may be used to enhance traffic management and planning.

# Chapter 5- Conclusion and Future work

## 5.1 Conclusion

In conclusion, the use of machine learning models to predict traffic congestion has considerable promise for enhancing transportation management and decreasing traffic-related problems. This experiment proved the method's viability by demonstrating its ability to predict traffic congestion with high accuracy using a combination of parameters and dataset analysis. Predictive rigour and accuracy are enhanced through thorough data preprocessing, exploratory data analysis, and the evaluation of multiple machine learning models.

What we learn from this experiment is that good data and thorough preprocessing are crucial for reliable outcomes. Extraction of temporal characteristics and the use of feature engineering approaches was useful in identifying important relationships within the dataset. Insights into traffic congestion patterns, peak hours, and variations between days of the week and interchanges were also supplied by the exploratory data analysis. These findings can help with transportation planning, resource allocation, and overall decision making.

Multiple machine learning models' abilities to foresee traffic congestion were assessed and compared. When comparing various models' performance on regression metrics, the Gradient Boosting Regressor emerged as the clear victor. By fine-tuning the models' hyperparameters, we were able to significantly improve their performance and bring forth their full predictive potential.

Nonetheless, it's critical to note that the experiment included some significant caveats. The dataset may not be able to capture all of the elements driving traffic congestion, such as weather and real-time traffic data, due to its restricted capabilities. The model's predicted accuracy and robustness might be improved by investigating new data sources or amassing larger datasets. Additionally, the experiment only considered one set of machine learning models, although taking into account additional models or ensemble techniques would have expanded the scope of the research and enhanced the accuracy of the predictions.

Overall, this research contributes to the field of traffic congestion prediction by demonstrating the effectiveness of machine learning models and providing insights into traffic patterns and correlations. The findings can inform transportation authorities in their efforts to manage traffic congestion, optimize resource allocation, and improve overall traffic management strategies.

## 5.2 Research Contributions

In the areas of traffic congestion prediction and transportation management, this experiment makes several contributions. As a first step, the all-encompassing strategy for data preprocessing sheds light on why it's so crucial to have high-quality inputs for machine learning models. Data preprocessing is crucial for obtaining important information and enhancing model performance, as shown by the use of feature engineering approaches and the treatment of outliers.

Secondly, the exploratory data analysis carried out in this experiment sheds light on the causes and effects of traffic congestion. The dataset's visualisations and analyses show daily traffic volume changes and trends, illuminating peak hours and shedding light on how drivers behave at different times of the day. Additionally, the study reveals fluctuations in traffic volume across different days of the week, assisting transportation authorities with resource planning and allocation. Traffic volume by intersection analysis identifies intersections with higher levels of congestion, providing useful data for traffic management and planning.

The best algorithm for traffic congestion prediction can also be chosen with the help of evaluation and comparison of several machine learning models. Model predictiveness may be quantitatively evaluated with the help of performance metrics like MSE, RMSE, and MAE. Results show that the Gradient Boosting Regressor excels on this particular dataset, demonstrating its potential for precise traffic congestion forecasting.

## 5.3 Future Work and Developments

The area of traffic congestion forecasting has several potential future directions for research and advancement. To begin, the accuracy and robustness of the predictive models can be improved by including more important information such as weather conditions, road conditions, and real-time traffic data. These characteristics can help capture the ever-changing nature of traffic congestion and shed light on its root causes and recurring trends.

Second, the accuracy of traffic congestion prediction can be enhanced by investigating different machine learning models and ensemble methodologies. Neural networks, support vector regression, and time series models are just a few examples of the types of models that can provide novel insights and accurately model intricate interrelationships in the data. Combining the benefits of several models into one, through stacking or blending, can result in more accurate forecasts.

The results of this study can be verified and generalized by gathering larger and more varied datasets in future studies. This would require factoring in data from multiple areas, multiple times, and multiple traffic scenarios. The dynamics of traffic congestion could be better understood, and more precise predictions could be made with a more extensive dataset.

Additionally, dynamic models that incorporate real-time data streams can increase the responsiveness and adaptability of traffic congestion forecasting. More proactive traffic management and congestion reduction tactics are made possible by incorporating real-time traffic information into the models so that they can change their projections based on the current traffic circumstances.

## 5.4 Personal Reflections

This experiment has served as a great educational opportunity. It has shed light on the difficulties of congestion forecasting and the promise of machine learning methods in resolving transport management issues. The experiment showed that data preprocessing, exploratory data analysis, and model evaluation are all crucial to making reliable forecasts.

Problems were faced at every stage of the experiment, from dealing with outliers to picking the right models to fine-tuning hyperparameters. Since overcoming them necessitated exercising one's analytical, problem-solving, and decision-making chops, they also constituted valuable learning and development opportunities. Understanding the benefits and drawbacks of various models was enhanced via the process of assessing and comparing them.

Looking back at the experiment, it's clear that more work and study is needed to perfect the science of predicting traffic congestion. There is a pressing need for more research and development because of the stated constraints, such as the absence of certain features and the small number of available models. Improvements in data collecting, modelling methods, and real-time data integration are areas where future study should concentrate.

Overall, the results of this experiment have improved our capacity to foresee and plan for traffic congestion. It has laid the groundwork for future studies and innovations in the field, all of which will hopefully lead to more efficient transport systems with less congestion.

# References

Abdullah, S. M. *et al.* (2023) ‘Optimizing Traffic Flow in Smart Cities: Soft GRU-Based Recurrent Neural Networks for Enhanced Congestion Prediction Using Deep Learning’, *Sustainability (Switzerland)*, 15(7). doi: 10.3390/su15075949.

Adetiloye, T. and Awasthi, A. (2019) ‘Multimodal Big Data Fusion for Traffic Congestion Prediction’, *Multimodal Analytics for Next-Generation Big Data Technologies and Applications*, pp. 319–335. doi: 10.1007/978-3-319-97598-6\_13.

Anjaneyulu, M. and Kubendiran, M. (2023) ‘Short Term Traffic Flow Prediction Using Hybrid Deep Learning’, *Computers, Materials and Continua*, 75(1), pp. 1641–1656. doi: 10.32604/cmc.2023.035056.

Ata, A. *et al.* (2020) ‘Adaptive IoT Empowered Smart Road Traffic Congestion Control System Using Supervised Machine Learning Algorithm’, 00(00). doi: 10.1093/comjnl/bxz132.

Bai, M. *et al.* (2021) ‘PrePCT: Traffic congestion prediction in smart cities with relative position congestion tensor’, *Neurocomputing*, 444(xxxx), pp. 147–157. doi: 10.1016/j.neucom.2020.08.075.

Ban, X., Guo, C. and Li, G. (2016) ‘Application of Extreme Learning Machine on Large Scale Traffic Congestion Prediction’, 1, pp. 293–305. doi: 10.1007/978-3-319-28397-5\_23.

Cheng, W. *et al.* (2022) ‘Combination predicting model of traffic congestion index in weekdays based on LightGBM-GRU’, *Scientific Reports*, 12(1), pp. 1–13. doi: 10.1038/s41598-022-06975-1.

Devi, S. and Neetha, T. (2017) ‘Machine Learning based traffic congestion prediction in a IoT based Smart City’, pp. 3442–3445.

Elfar, A., Talebpour, A. and Mahmassani, H. S. (2018) ‘Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment’. doi: 10.1177/0361198118795010.

Gatto, R. C. and Forster, C. H. Q. (2021) ‘Audio-Based Machine Learning Model for Traffic Congestion Detection’, *IEEE Transactions on Intelligent Transportation Systems*, 22(11), pp. 7200–7207. doi: 10.1109/TITS.2020.3003111.

Impedovo, D. *et al.* (2019) ‘Vehicular traffic congestion classification by visual features and deep learning approaches: A comparison’, *Sensors (Switzerland)*, 19(23). doi: 10.3390/s19235213.

Julio, N., Giesen, R. and Lizana, P. (2016) ‘Real-time prediction of bus travel speeds using traffic shockwaves and machine learning algorithms’, *Research in Transportation Economics*, 59, pp. 250–257. doi: 10.1016/j.retrec.2016.07.019.

Khan, M. A. *et al.* (2019) ‘MODELLING SMART ROAD TRAFFIC CONGESTION CONTROL SYSTEM USING MODELLING SMART ROAD TRAFFIC CONGESTION CONTROL SYSTEM USING Introduction’, (May). doi: 10.14311/NNW.2019.29.008.

Kothai, G. *et al.* (2021) ‘Research Article A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities’, 2021.

Kumar, S. A. P. *et al.* (2018) ‘A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance’, *Journal of Reliable Intelligent Environments*, 4(4), pp. 199–209. doi: 10.1007/s40860-018-0069-y.

Li, L. *et al.* (2020) ‘MF-TCPV : A Machine Learning and Fuzzy Comprehensive Evaluation-Based Framework for Traffic Congestion Prediction and Visualization’, 8, pp. 227113–227125. doi: 10.1109/ACCESS.2020.3043582.

Li, T. *et al.* (no date) ‘Short-term traffic congestion prediction with Conv – BiLSTM considering spatio-temporal features’. doi: 10.1049/iet-its.2020.0406.

Li, Z. *et al.* (2017) ‘Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks’, *IEEE Transactions on Intelligent Transportation Systems*, 18(11), pp. 3204–3217. doi: 10.1109/TITS.2017.2687620.

Liu, Y. and Wu, H. (2017) ‘Prediction of Road Traffic Congestion Based on Random Forest’, pp. 361–364. doi: 10.1109/ISCID.2017.216.

Ma, X. *et al.* (2015) ‘Large-scale transportation network congestion evolution prediction using deep learning theory’, *PLoS ONE*, 10(3), pp. 1–17. doi: 10.1371/journal.pone.0119044.

Martínez-álvarez, G. A. E. F. A. T. F. (2016) ‘A novel methodology to predict urban traffic congestion’, *Soft Computing*. doi: 10.1007/s00500-016-2288-6.

Meena, G. (2020) ‘Traffic Prediction for Intelligent Transportation System using Machine Learning’, (February), pp. 7–8.

Method, F. L. (2019) ‘SS symmetry Large-Scale Tra ffi c Congestion Prediction Based on the Symmetric Extreme Learning Machine Cluster’.

Mihaita, A.-S., Li, H. and Rizoiu, M.-A. (2020) ‘Traffic congestion anomaly detection and prediction using deep learning’, pp. 1–25. Available at: http://arxiv.org/abs/2006.13215.

Moridpour, S. (2021) ‘A Review of Traffic Congestion Prediction Using’, 2021.

Navarro-Espinoza, A. *et al.* (2022) ‘Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms’, *Technologies*, 10(1), pp. 1–11. doi: 10.3390/technologies10010005.

Othman, M. S. Bin, Keoh, S. L. and Tan, G. (2017) ‘Efficient journey planning and congestion prediction through deep learning’, *2017 International Smart Cities Conference, ISC2 2017*. doi: 10.1109/ISC2.2017.8090805.

Park, J. *et al.* (2009) ‘Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion’, 58(9), pp. 4741–4756.

Qi, Y. and Cheng, Z. (2023) ‘Research on Traffic Congestion Forecast Based on Deep Learning’, *Information (Switzerland)*, 14(2). doi: 10.3390/info14020108.

Razali, N. A. M. *et al.* (2021) ‘Gap, techniques and evaluation: traffic flow prediction using machine learning and deep learning’, *Journal of Big Data*, 8(1). doi: 10.1186/s40537-021-00542-7.

Reshma Ramchandra, N. and Rajabhushanam, C. (2021) ‘Machine learning algorithms performance evaluation in traffic flow prediction’, *Materials Today: Proceedings*, 51(xxxx), pp. 1046–1050. doi: 10.1016/j.matpr.2021.07.087.

Saleem, M. *et al.* (2022) ‘Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques’, *Egyptian Informatics Journal*, 23(3), pp. 417–426. doi: 10.1016/j.eij.2022.03.003.

Shen, Z. and Key, B. (2020) ‘Traffic Congestion Prediction using Decision Tree , Logistic Regression and Traffic Congestion Prediction using Decision Tree , Logistic Regression and Traffic Congestion Prediction using Decision Tree , Logistic Regression and Traffic Congestion Predicti’, *IFAC PapersOnLine*, 53(5), pp. 512–517. doi: 10.1016/j.ifacol.2021.04.138.

Sun, S., Chen, J. and Sun, J. (2019) ‘Traffic congestion prediction based on GPS trajectory data’, *International Journal of Distributed Sensor Networks*, 15(5). doi: 10.1177/1550147719847440.

Sunindyo, W. D. and Satria, A. S. M. (2020) ‘Traffic congestion prediction using multi-layer perceptrons and long short-term memory’, *EECCIS 2020 - 2020 10th Electrical Power, Electronics, Communications, Controls, and Informatics Seminar*, pp. 209–212. doi: 10.1109/EECCIS49483.2020.9263483.

Walraven, E., Spaan, M. T. J. and Bakker, B. (2016) ‘Traffic flow optimization: A reinforcement learning approach’, *Engineering Applications of Artificial Intelligence*, 52, pp. 203–212. doi: 10.1016/j.engappai.2016.01.001.

Wang, P. *et al.* (2018) ‘Regional Detection of Traffic Congestion Using in a Large-Scale Surveillance System via Deep Residual TrafficNet’, *IEEE Access*, 6, pp. 68910–68919. doi: 10.1109/ACCESS.2018.2879809.

Yasir, R. M., Nower, D. N. and Shoyaib, D. M. (2022) ‘Traffic Congestion Prediction Using Machine Learning Techniques’. Available at: http://arxiv.org/abs/2206.10983.

fedesoriano (no date) “Traffic Prediction Dataset.” Available at: https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset

de Harder, H. (2023) *Ethical considerations in machine learning projects*, *Towards Data Science*. Available at: https://towardsdatascience.com/ethical-considerations-in-machine-learning-projects-e17cb283e072

Samuel, N. (2022) *Data Quality Analysis Simplified: A Comprehensive Guide 101*, *Learn | Hevo*. Available at: https://hevodata.com/learn/data-quality-analysis/

Awan-Ur-Rahman (2019) *What is data Cleaning? How to process data for Analytics and Machine Learning modeling?*, *Towards Data Science*. Available at: https://towardsdatascience.com/what-is-data-cleaning-how-to-process-data-for-analytics-and-machine-learning-modeling-c2afcf4fbf45

Swaminathan, S. (2018) *Linear regression — detailed view*, *Towards Data Science*. Available at: https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86

Raj, A. (2020) *A quick and dirty guide to random forest regression*, *Towards Data Science*. Available at: https://towardsdatascience.com/a-quick-and-dirty-guide-to-random-forest-regression-52ca0af157f8

Masui, T. (2022) *All you need to know about gradient boosting algorithm − part 1. Regression*, *Towards Data Science*. Available at: https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502 (Accessed: July 17, 2023).

Chelliah, I. (2020) *An introduction to K-nearest neighbors algorithm*, *Towards Data Science*. Available at: https://towardsdatascience.com/an-introduction-to-k-nearest-neighbours-algorithm-3ddc99883acd (Accessed: July 17, 2023).

Brownlee, J. (2020) *Stacking Ensemble Machine Learning With Python*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/ (Accessed: July 17, 2023).

Brownlee, J. (2020) *Blending Ensemble Machine Learning With Python*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/blending-ensemble-machine-learning-with-python/ (Accessed: July 17, 2023).

Lee, W.-M. (2021) *Tuning the Hyperparameters of your Machine Learning Model using GridSearchCV*, *Towards Data Science*. Available at: https://towardsdatascience.com/tuning-the-hyperparameters-of-your-machine-learning-model-using-gridsearchcv-7fc2bb76ff27 (Accessed: July 17, 2023).

Raj, R. (no date) *Evaluation Metrics for Regression Models*, *Enjoyalgorithms.com*. Available at: https://www.enjoyalgorithms.com/blog/evaluation-metrics-regression-models (Accessed: July 17, 2023).