2024

Intelligent Traffic Management Systems

Mohammad Mazhar
Minnesota State University, Mankato

Intelligent Traffic Management Systems

By

(Mohammad Mazhar)

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

(Master of Science)

In

(Data Science)

Minnesota State University, Mankato

Mankato, Minnesota

(April 2024)

April 12, 2024

Intelligent Traffic Management Systems

Mohammad Mazhar

This thesis has been examined and approved by the following members of the student's committee.

____
Dr. Suboh Alkhushayni

____

____

Acknowledgments

I am profoundly grateful to my dedicated and insightful supervisor, Dr. Suboh Alkhushayni, whose exceptional guidance, unfaltering support, and remarkable patience have been indispensable throughout the entirety of my master's degree pursuit. Dr. Suboh Alkhushayni profound expertise and extensive experience have not only shaped the intellectual landscape of my academic journey but have also been a constant source of inspiration and motivation.

In addition to providing invaluable insights into the intricacies of my research, Dr. Suboh Alkhushayni mentorship has fostered an environment of intellectual curiosity and academic rigor. Their unwavering commitment to my academic and professional development has been instrumental in navigating the challenges and complexities of my research endeavors. I extend my heartfelt appreciation for the countless hours spent in discussions, the constructive feedback provided, and the enduring encouragement that has fortified my academic resilience.

Table of Contents

Intelligent Traffic Management Systems

Abstract

With the increase in population and in particular urban population. The traffic and travel times in between cities and inside cities has increased due to more and more people using private means of transportation. Due to this need arose for tackling the increase in traffic by managing it using various means. For this we look towards The Intelligent Traffic Management System (ITMS). ITMS is an AI-powered solution designed to optimize traffic flow, reduce congestion, and improve overall road safety. The system will monitor real-time traffic data using a combination of cameras and sensors, identify traffic jams, and send alerts to traffic authorities and police officers. This paper reviews the key components of ITMS. Which includes traffic flow prediction, image classification and incident or accident detection through cameras, sensors, and inductive loop detectors. Comparison and evaluation metrics of different algorithms and models are presented. Tables show the accuracy rates and best features of various techniques. After that future work is presented in which a framework is presented to write and combine the best practices of each component together to create a useful ITMS. Which can be used in future smart cities, big urban centers and could also be used as a guide.
Keywords: *Intelligent Traffic management systems, inductive loop detectors, optimize traffic flow, reduce congestion*

CHAPTER ONE

INTRODUCTION

1.1 Background

Transportation is a key player in our human lives today around the globe. Various means of transportation are used for different kind of purposes. Airplanes are widely but not always are used for travelling extremely long distances quickly. Which has made it easier to reach destinations in hours which used to take travelers days, weeks and sometime even months. Due to this reason the world has become increasingly globalized. Other factors like internet, mainstream and now social media have increased this trend even more so. Apart from airplanes we use bicycles and motor bike for travelling shorter distances. According to world economic forum today 76% of Americans opt for car as their means of transportation to travel between home and work. Only 11% use bike. In Germany 65% of people use their car to travel from home to work. And in Netherlands a country which is famous for its clean environment and using bike to commute also opts for car as means of transportation over bike 56% to 36%.

This makes it obvious that all around the world cars are the most used means of transportation. There is plethora of advantages for using your own transport. Like reduced travel time, less wait time, less costly in most cases and more comfortable etc. But there are disadvantages as well for like traffic congestions, more frequent accidents, increased costs, environmental and ecological, fuel and other resources being drained. Maintenance costs and in snowing areas tires and battery costs increase. This naturally makes us want to find solutions for these problems and look to make transportation easier, reliable, smart, and intelligent.

Enter Intelligent Transportation Systems (ITMS) it is an approach which looks to solve the problems associated with transportation and take it to next level by integrating it with technology. It is not limited to one specific domain of vehicle or airplane transport system. It is a broad field that includes multiple other fields. (Naseer and Abdullah 2013) showed in their extensive survey on Intelligent Transportation Systems (ITS). The domains that are included in this one field of study are as follows:

1) Intelligent Traffic Systems
2) Vehicle to Vehicle ad hoc networks
3) Incident management systems
4) Freeway management systems
5) Freight management systems
6) Emergency management systems

These are few of the main fields in ITMS that have recently attracted a lot of attention. As the field gets more researchers involved and new methods are invented for transport or current ones are enhanced. This area will get bigger and more sophisticated. Out of all these subfields the two fields that have attracted the most researchers and the most literature dedicated towards is (1) Intelligent Traffic Systems (2) Incident management systems.

Intelligent traffic systems are a concept for future smart cities and modern urban centers with big populations. That seeks to solve the problem of traffic flow, image classification in crowdy areas and traffic jams and seeks out a way to predict congestion by making it less congested. This all is done by making new algorithms and artificial intelligence techniques. Which are then put together with hardware and infrastructure to solve these problems of current traffic systems.

The other one is incident management systems which comes after incident management systems. There have been studies recently showing the improvement and advancements in incident management systems. The main purpose of these systems is to make the road travel safer. Because the most used mean of transportation is by cars or through road networks. Apart from making it safer it looks to make the emergency timing and alarming of incidents to the concerned authorities quicker and more accurate. This involves a hybrid approach (Parkanay and Xie 2005) classified the incident detection in two categories (1) Automatic incident detection (2) Non automatic incident detection.

(Hireche and Adeselem 2020) was showed in their review of AIDS that hybrid approach works best for the incident detections. We will review all main algorithms and find the most best AI techniques to enhance this system. Because by making roads safer and travel more efficient is the only way to tackle problem like road accidents, energy, fuel consumption and traffic congestion. Which creates problems not only for the environment and puts extra burden on traffic authorities. But it also leaves people annoyed, frustrated and angry by making traveling times longer. Both these subfields of ITMS are interconnected. If we propose a few solutions in the realm of ITS. Then most certainly many hurdles in incident detection systems can be jumped over as well. If one looks on IDS then one sees how reducing extra travel time which are caused by planned or unplanned events shown in figure

1 (Nikolaev and Sapego 2017) on the roads will help in reducing the time requires for policemen, emergency vehicles like ambulances, fire brigade and other rescue operations.

Figure 1. Type of incidents

American Transportation Research Institute estimates that the freight sector lost $74.1 billion a year due to congestion. Not only that American drivers lose a total of $88 billion a year due to excess traffic congestions which amounts to $1377 for each driver (Kyle 2023). Other economic costs are noise pollution, frustration and road rage by drivers, delay of goods and services, other related but hidden costs. This is the area which the application of ITMS can be made to take up the problem of traffic congestion. This begs the question of how and where to start. Answer lays in traffic flow prediction and image classification and processing.

The idea of smart cities is resting upon a lot of components out which one of the most important one is the intelligent transportation systems (ITS). This studies focal point intelligent traffic management system (ITMS) is a key component of ITS. Making it an integral and important part in the smart cities. A lot of research has been done on either ITMS directly or in the closely related concepts. But still a lot is yet to be done to make and take the ITS to a level that visibly starts to improve the living standards. Although the future of smart city is exciting with a lot of research already in place or in progress. It has been criticized by (Anthopoulos 2016) for its unfulfilled promises.

The reason why traffic becomes a challenge is not only the direct economic losses. But the emergency vehicles like police vans, ambulances, fire brigades and other rescue operation facilities suffer from this. (Amaral 2016) and authors show us in Rio de Janeiro real time data from thirty different agencies is drawn together, insights are pulled from that data and whole city is managed according to that. When it comes to solving the problem of traffic management and incident management. Developing new systems and integrating it with existing technology and assets that are owned by cities states alone won't make it easier to solve. For this researcher form now onwards should present a framework in which these systems can be deployed. Then form there it can be used by the authorities to make better decisions.

The reason why the framework becomes a necessity is that it allows the city and in particular concerned traffic authorities to see the bigger picture. As mentioned above how (Anthopoulos 2016) wrote that ITS even after having so much researcher's attention drawn into it and making progress it still underdelivers. One reason for this which can be seen right away is that only making algorithms work and developing new artificial intelligence-based techniques isn't it self-going to make the problems go away. The studies recently and even more so in past keep on overlooking the hardware parts of these systems. The technology required by these sophisticated models to be integrated within. And is it even possible for these algorithms/models to be planted into the current assets owned by the authorities in modern urban centers or not? these and many more questions arise once anyone takes up a challenge as big as solving the modern traffic and incident management.

ITMS can be seen removing a lot of these obstacles in making travel safer, assuring more efficient use of fuel and energy consumption, detecting incidents, and fixing them quickly by timely rescue operations and boost overall economic activities as transport is its backbone.

1.2. Objectives

The primary objectives of ITMS are:

- Monitor and analyze real-time traffic data to identify congested areas and potential bottlenecks.
- Predict traffic flow patterns using AI-driven forecasting models.
- Optimize traffic management strategies based on AI-generated insights.
- Provide real-time alerts and notifications to traffic authorities and police officers.
- Offer a user-friendly interface for traffic authorities to monitor and manage traffic conditions.
- Integrate the ITMS with existing traffic management infrastructure and systems.
- Identify best database strategies for timely efficient and quick analysis.

Apart from the objective that fall under ITMS. Another important part that needs to be focused on in transportation systems is Incident detection in urban cities and future smart cities. As both are interconnected.

Monitoring and analyzing of real time traffic data lies in the core of ITS. Because congestion creates most of the challenges face by the authorities, drivers, citizens, and all type of emergency services. How does the analyzing of real time traffic data takes place? By the algorithms and AI techniques that have taken over all the fields now.

Be it mathematics, biology, chemistry, or other natural sciences all researchers and experts in these fields are taking these fields to a next level by bringing new technologies. Social Sciences just like natural sciences is bringing new software's powered by AI and other sophisticated methods that enhance the capabilities of the scientists even in societal domain. Time series analysis in the fields of econometrics, psychometrics and sociometric are the proofs of how technology brings previously unthinkable research ideas into reality. These time series models can be leveraged to solve the prediction of traffic flow problem. The real time traffic data is similar in structure and database techniques are also same. Mathematics and statistics serve as a bridge in all the fields. Which in our case will help us in traffic flow prediction.

The applications made now a days have made it easier for us to order food online, deliver cloths and revive them online. Select which places to visit by checking their reviews. We can get real time notifications of sports, news, and weather conditions by checking these applications. These systems can be taken to next level by powering them with AI. Though which real time traffic data, most cost-efficient travel time can be calculated. It can also help in increasing the response time of traffic authority's emergency and rescue operation services by helping them get exact and accurate information by these systems.

The interface needed for all these authorities, management, and administrations to work with should be easy to use and user friendly. For instance, a software engineer cannot be expected to sit on the seat of a traffic policemen and do his work. Just like that other way around is the same. The interface that why needed for ITMS must be user friendly as such it helps user rather than making them more confused when they use it. The integration of all the algorithms and models in these apps must be correctly done whether its data analytics, image classification or traffic routes recommendation after predicting the traffic flow.

1.2.1 Hardware:

The hardware components of the ITMS include:

- Traffic cameras: High-definition cameras strategically placed at intersections, highways, main roads, and other high-traffic areas to capture live video feeds.
- Sensors: Inductive loop detectors, radar, and infrared sensors installed on roads to collect data on traffic volume, speed, and occupancy.
- Central server: A high-performance server responsible for data storage, processing, and running AI models.

1.2.2 Software:

The software components of the ITMS include:

- Traffic Monitoring Application: A software application responsible for processing and displaying live traffic data from cameras and sensors.
- Data Processing and Storage: A set of tools and libraries for processing and storing traffic data, ensuring consistency and reliability.
- AI Model Training and Deployment: A framework for training, deploying, and managing AI models used for traffic prediction and optimization.

1.2.3 AI Models and Algorithms:

The AI models and algorithms employed in the ITMS include:

- Object Detection: Models such as ANN or CNN for vehicle detection and tracking in traffic camera feeds.
- Traffic Flow Prediction: Time-series forecasting models like LSTM or Prophet for predicting traffic flow based on historical data.
- Traffic Management Optimization: Reinforcement learning models like Deep Q-Network or Proximal Policy Optimization for determining optimal traffic management strategies.

It is important to keep the cost of these hardware and software components in mind as well. The best case will be where they can be easily integrated with the existing infrastructure. If not, then most necessary assets are to be recommended to the concerned authorities which we will be discussed in chapter 5.

1.3 Contributions per chapter

In chapter 2 we will do a comprehensive literature review of traffic flow prediction models and how the models used have evolved overtime. Then we will discuss and analyze the key points from that comprehensive literature review. After that we relevant studies done on image classification, incident detection, and traffic congestion algorithms will be reviewed.

In chapter 3 we will focus on methodology and deployment of models on real world datasets. In which key statistical, machine learning, and deep learning models will be applied on these selected datasets for traffic flow prediction. After this is done for image classification purposes, we will use built in TensorFlow dataset. Which will be used for the image classification part of the ITMS. Then for intelligent incident detection and traffic congestion algorithms we will recommend the most accurate models and algorithms discussed in literature for both.

In chapter 4 we will discuss the database techniques and the important databases need to record and analyze data for real time deployment of these models. This is for making it easier for relevant authorities to use these databased for real time use. The key functionalities of GUI will also be addressed.

In chapter 5 we discuss the hardware and software components of ITMS. The steps needed to bring these components for real time deployment. The methods to be used for making process efficient overall. The techniques and type of activities needed for the process like agile and scrum.

In chapter 6 we conclude the entire ITMS. The key parts of the literature reviewed the types of models used and the one's recommended form the existing literature. The recommendations for future work will also be discussed. In which future developments for data sources, public private partnership, and institutional knowledge exchange is discussed.

The purpose of this study is to propose a comprehensive framework for intelligent traffic management systems. Ideas and systems existing in this field currently are old and miss new AI methods and algorithms. That's why this study focuses on incorporating new AI and database techniques by applying them on real world and real time datasets. Hardware, software components as well as project management methods will also be included. Because the goal is to make management system for intelligent traffic in future smart cities and modern urban cities.

CHAPTER Two

Literature Review

2.1 Traffic Flow Forecasting

Although the researchers have been for a long time trying to perfect the modes that forecast traffic flow. But due to the entry of deep learning models the field has changed significantly. The accuracy of these time series forecasting models tend to be higher than average statistical and machine learning models. That is why it is important to look at the studies done individually that use these models. This will give us better insight of these models and how over the period models have changed and improved. We can classify their types of models to be reviewed separately.

1. Statistical Models.
2. Machine Learning models
3. Deep learning models

2..1.1 Statistical Models

(Jiu and Guan 2004) summarized all the model up until their research. And showed that historical average model (HA) and Autoregressive integrated moving average (ARIMA) were used initially in this field to forecast traffic. This was one of the first surveys of the traffic flow forecasting methods. Almost all the models reviewed here were statistical model or their extensions. Stephanedes forecasted traffic by using the HA model back in 1981. This model takes the average of data of prior periods and then tries to predict ahead. The accuracy however is relatively low for this model. But the time that it takes to set up and calculate is lower than others. One of the most used time series models in past fifty years is ARIMA. Which was proposed by Box in 1976. This model includes Autoregressive (AR) and Moving average (MA) the integration mean it differences the sequence of current time step from the previous one.

Historical Average and ARIMA are both statistical models. (Ahmed and Cook 1979) also applied ARIMA to Historical Data. This became one the first works. This was used to predict traffic flow on the freeways. This was just three years after (Box and Jenkins 1976) proposed and extensive ARIMA model in their study. In the previous extension of their research (Stephanedes and Okutani 1984) applied a Kalman filter model to the Minneapolis and Nagoya city to predict the urban traffic flow. Because the Time series of any specific data captured show periodic variation. Whether it is weekly, monthly, or quarterly. It makes it almost a necessity to capture these variations. SARIMA which stands for Seasonal Autoregressive Integrated Moving Average fills in that gap. It is an extension of basic statistical ARIMA.

(Gosh and Basu 2005) in another one of their studies showed a time series model by applying SARIMA on Dublin, Ireland dataset. That seasonal adjustment yields better result. (Gosh and Basu 2009) did an extensive analysis on multivariate times series analysis for short term traffic flow forecasting. They developed a 'multivariate structured time series model' in multi variate form. Which proved to be accurate and effective in urban traffic flow prediction. The model's approach was to not develop new method for forecasting from scratch. But instead of choosing flow, occupancy or speed the way usually univariate forecasting is taken. They took input from different routes and various sites. Which made the model multivariate. This helps in checking the dependency of various traffic roads or sites on each other. And helps answer important questions like how much traffic from x direction flows to y? And how to much on average the speed changes going from x route to y? This gives vital input in managing traffic operation in short run.

(William and Hoel 2005) in their analysis showed that seasonal ARIMA predicts with better accuracy in univariate forecasting. They proved it by applying one week's difference in their model. Although their model was applied to only one variable but proved the point of SARIMA giving a better result ARIMA. (Wang et. al 2008) Did a study which extended the baseline Kalman filtering model and stochastic macroscopic model to predict the real time freeway traffic. (Li 2016) predicted the traffic flow using the interval type-2 fuzzy sets theory.

(De Voort et. Al 1996) did a study in which a hybrid model was made and used to predict the short-term traffic flow on the French highway. The author named it as the "KARIMA" model which uses the kohonen map which basically has hexagonal structure making it easier to define the classes within the data. When separated explicitly the classification and approximation of functions greatly improves the accuracy of KARIMA model compared to ARIMA model or even backpropagation neural network. The model was tested on time series which had half an hour- and hour-long breaks. Performance in the end is very similar to the one that other layered models give. The advantage the model showed was that it uses only two to four classes. This concludes that since the number of classes s small so it can be used in the prediction of long-term traffic as well.

(Lee and Fambro 1999) in their research for traffic planning purposes applied a subset ARIMA model for short term traffic volume forecasting. The identification for the model was done using Akaike's info criterion. Conditional likelihood estimation method was used for estimating the parameters. For the verification of white noise process was used twice. To analyze the results, four different types of models were used. In this traffic was to be predicted one step ahead of current time step. Two different types of metrics were used to measure the accuracy of the time series models. The results in the end showed that although full ARIMA model gives good and reasonable accuracy. The subset ARIMA gives higher accuracy compared to it.

(Schimbinschi et. Al 2017) made a very important contribution by making a new model based on extension of Vector Autoregression called Topology Regularized Universal VAR (TRU-VAR). It is already a known fact that baseline VAR is one the most widely used multivariate timeseries models. This enhancement of the baseline will help increase the prediction accuracy of the model. There were two datasets used here by the authors both were state of the art real time traffic datasets in large urban centers. First one from Melbourne and second from California freeway traffic measurement. In both cases the enhancement of baseline VAR outperformed the ARIMA model.

(William 2001) also showed in his study how even the extension of baseline ARIMA called ARIMAX tends to outperform ARIMA. The data used in this comparison was from upstream traffic. The cost here was that ARIMAX is computationally more complex and thus more time consuming than ARIMA. (Xie et. Al 2007) in their study compare two types of Kalman filtering. The baseline Kalman filtering methods and wavelet Kalman filtering. The results showed that wavelet Kalman filtering method outperforms the baseline in mean absolute percentage error and root mean square error.

(Kumar and Vanajakhsi 2015) also did a study on the data from a large urban center Chennai and compare two ARIMA and SARIMA. The data used here was a 3-lane arterial routes in the city. The study also compares the historical average methods. The results in the end show that SARIMA gives a higher accuracy, but the computational time might be higher than the other models.

2.1.2 Machine Learning models

With the advent of advanced statistical methods and the culmination of computer processing power which exceeds far more than anything a human can do itself. Gives rise to machine learning and gives the time series modeling computations a whole new twist for the better. Because it exceeds not only in the data processing capabilities but also the multiple variables and memory. These both factors put together leads to the rise of machine learning methods which are more advanced and helpful in time series modeling then statistical models. This gives us opportunity in our field to be applied to forecast traffic flow.

(Clark 2003) did one of the first studies in the field of machine leaning that was better than the baseline statistical models. This was also unique in a sense that it had an on-site accurate prediction ability which not a lot of research papers offer. This focuses on the three-dimensional nature of the traffic state. It is basically using an intuitive method of pattern matching to predict the flow. It is also one of the first attempts to forecast multivariate traffic prediction which is based on non-parametric regression. The data used in the study was from London motorway. The results showed promising accuracy and can be used in prediction for real time traffic prediction.

(Smith and Demestky 1996) was another one of the earlier works done in the field of non-parametric regression which moves beyond normal statistical models. This model developed forecasts traffic in multiple intervals on freeways. The goal was to predict the traffic volume for coming several hours in 15 minutes intervals. This is a generic algorithm which can be used in real time field locations for forecasting. The data used was from Northern Virginia Traffic management authorities and is applied to two sites on the Capital Beltway. In both cases short term and long-term forecasting, the proposed model gave accurate estimates.

Initially in the models developed in machine learning didn't give results higher enough as compared to statistical method but computational complexity made it easier for ARIMA or SARIMA to be used. (Lee et. Al 2000) which compared a non-parametric regression technique with classical SARIMA. The non-parametric technique used here is the k nearest neighbor with adjusted weights. The results of these were like the ones produced by the SARIMA. The models in machine learning also took time to be perfected and the above-mentioned studies were that why a little unique in this sense.

(Lv et. Al 2009) also applied knn's to predict highway traffic accident detection. This was also the first time any such study was done in history of machine learning literature. Before the prediction of accidents precursor and their time slice duration are first identified. The results showed that the knn does better job than C-mean clustering which were the traditional way of accident prediction. (Hogberg 1976) became one of the earliest works that can be resonated with non-linear regression approaches. The attempt was not to develop any advanced machine learning model. Because even ARIMA model itself had been the same year advancement. But the study certainly shows how early in this field the idea nonlinearity was being put forth.

(Kim et. Al 2005) made a key contribution as well in their study by showing how memory less property in traffic forecasting leads to problematic results. The current nearest neighbor nonparametric forecasting traffic forecasting models treat evolution of dynamic flows as a memoryless process. Which means that whichever traffic we see on the road currently is independent of past. This attempt was to look at the past sequences of traffic and then predict future based on those. The results showed that the proposed model of past sequential analysis and prediction based on those does a better in forecasting flow then non-parametric Knn.

(Li et. Al 2012) proposed a model which extended the baseline Knn model by locally weighting it first and then regressing it. In this way the traffic that is flowing with the same speed are considered as neighbors. And the predicted values which is the result of prediction is now as nearest neighbors. Every nearest neighbor i.e. predicted value has different similarity with the original value. This similarity must be maximized. The model was tested on real time collected data from two sources in China. Results show that K-LWR improves the accuracy significantly over the baseline model KNN.

(Castro-Neto et. Al 2009) presents a different approach in their research. Instead of forecasting traffic with typical conditions where there is no incident no hurdles in the way. They take an atypical approach. In this all the conditions like holiday, roads closure due to accident or other various reasons are added. They called the model Online Support Vector Regression OL-SVR. This way the prediction will be more realistic. They compared the model with three well-known prediction models Gaussian maximum likelihood (GML), Holt Exponential Smoothing and Artificial Neural Networks (ANN) models. Results show that GML does a slightly better job in typical traffic prediction. But the OL-SVR is best performer among all models in atypical forecasting.

(Chiang Hong et. Al 2011) developed a model which extends the baseline SVR model to predict non-linear time series predictions. They made this model for short term traffic data prediction that has non-linear patterns. Continuous ant colony optimization algorithm is added into the SVR mode for forecasting. For testing purposes data is used from northern Taiwan. The results of the model show that SVRCACO model yields better results than SARIMA. Therefore, making at a better time series forecasting model.

(Li et. Al 2012) made another notable contribution by proposing a weighted pattern recognition algorithm (WPRA). Knn non-parametric regression is a classical model for a single point traffic flow forecasting. When traffic flows in the same of the day that is considered as neighbors and neighbors with the closest values to each other are regarded as nearest neighbors. After an observation one can notice that trends in the traffic flow are important. And it gives rooms to studying those past sequences and then predicting traffic based off that. For example. If certain amount of traffic passes through one point in time in the evening as morning. The concerned authorities can look at the past values of previous days traffic in the evening to predict traffic rather than morning values. As the evening values are more relevant. The results show that WPRA improves the accuracy significantly over PRA.

(Hong et. Al 2015) made an important contribution by proposing a hybrid multi metric knn model (HMMKMM). The classical Euclidean distance models based on metrics are not effective because they give equal weight to every feature in the multi-source high dimensional feature space. The second thing this study focuses on is improving the traditional way of making models by experts which are error prone. The third being the complex nature of traffic in real time. This model attempts to analyze the intrinsic features of data and reduce the semantic gap between domain knowledge and feature engineering that is done by human sources. Results show that HMMKNN improve the results and gives better accuracy than traditional Euclidean based KNN.

(Wu et. Al 2004) took a different approach at traffic flow prediction instead of predicting flow directly they tried to predict the time travel instead. Because in the broader field of ITS it is of vital importance. They applied the baseline SVR model to the rea time traffic data from Taiwan highway. Then the model in the end was compared to the historical mean model for time series prediction. Results show that SVR outperforms the historical mean baseline model and gives higher accuracy.

(Zheng el. Al 2008) developed a model called accurate online SVR (AOSVR) which included artificial neural networks as well. The reason for combining the ANN here was to increase the time efficiency as well not only the accuracy of traffic flow forecasted. The traffic data used for the study was from Guangzhou, China for twelve hours of the day with intervals every five minutes. The results of show that AOSVR with ANN increase the accuracy of model and is efficient as to be used in real tie traffic data analysis.

(Asif et. Al 2013) also made a notable contribution by making a model to identify spatio temporal patterns for large scale traffic speed prediction. Normally when traffic tends to have patterns when forecasted on those patterns it gives good accuracy but leaves room for improvement. That's why studying these spatiotemporal patterns can improve the results even further. In this research the authors used k means cluster, principal component analysis and self-organizing maps put all these together to mine spatiotemporal trends for individual links between the networks. The results show that SVR does a better job than other time series forecasting models.

(Ahnn et. Al 2015) in their research combined two most widely used time series prediction models like Bayesian classifiers and SVR. The data used was real time traffic data. That's why first 3-D Markov model was used to model the spatiotemporal domain of the data. After that based on the results cliques of current cone zone and its neighbors were identified. The estimation of the cliques defined was made based on multiple linear regression and SVR. The data used was from Gyeongbu expressway. Results show that approach that uses SVR based estimation is superior in accuracy than linear based regression.

(Yao et. Al 2020) made one of the more recent notable contributions in this field. Here two methods one linear and the no linear are compared for short term traffic prediction. The nonlinear method here can be classified as hybrid because it combines Markov model with the wavelet neural network. ARIMA-GARCH model is used here to predict the more volatile rends in the data. And other method focuses on the prediction of irregular parts. The data used is from China. Results show that hybrid nonlinear prediction method gives higher accuracy. It was able to predict the intervals more accurately than linear method.

(Huang et. Al 2022) took a different approach in prediction for short term traffic. The time series here was decomposed TSD. Flow of traffic was decomposed into three parts periodic, residual, and volatile component. To extract the intrinsic mode functions empirical mode decomposition is applied. Then Hilbert transformation is applied and presented in terms of frequencies. It then converts the data into Fourier series. IMF presents the lowest frequencies of residual part. The volatility component is modeled by supervised learning. Results show that TSD can discover hidden periodic patterns. Volatile component can predict the accuracy of next step with high accuracy.

(Feng et. Al 2019) The researchers here take a different approach their goal in the end is to minimize traffic congestion in the long run. That's why they predict the traffic flow in adaptive way for on point real time prediction. The algorithm developed is named adaptive multi-kernel SVM (AMSVM). First is to account for the nonlinearity in data. Second step is to optimize the parameters by applying the particle swarm optimization algorithms. To detect the changing tendencies of traffic flow. Third part is to add spatiotemporal correlation in AMSVM. Results show that AMSVM outperform other models for forecasting. The method adapts quickly during the rush hour and accurately predicts the flow.

(Tian et al. 2021) The researchers here made a spatiotemporal feature vector model combined with the random forest (SFRF). First the preprocessing algorithms are studied and applied for data cleaning and analysis purposes. This way the Electronic Toll Control (ETC) data can be understood. Spatiotemporal vector model can then be constructed based on this information from highway. The developed method is then used on the expressway data from the Fujian province. Then the comparison is made with three commonly used baseline statistical and machine learning models HA, LR, and GBDT. Results show that developed method of traffic flow forecasting gives the highest accuracy when compared with other three model.

(Jeong and Byon 2013) in their study developed an algorithm called online weighted support vector regression (OLWSVR). In this approach unlike the classical linear regression the regression parameters are increased or decreased step by step whenever a new data is added. The method in their study takes two samples at first step. After which it generates OLWSVR coefficients. Once that is done algorithm moves ahead takes another step ahead by taking three samples next time and train the remaining part.

If the new sample is not support vector, then it is added without any updates. But if the vector is found to be support vector. Then it is incrementally updated. This is how the online-based SVR is put together with weighted learning method. And then applied a California real time dataset which takes data from 30 loop detectors. Dataset is benchmark and is named California Freeway Performance Measurement. It is available for research purposes for anyone to use. It basically aggregates data every 5 minutes. The researchers came up with two different scenarios. Results showed how in both scenarios the mean absolute percentage error (MAPE) was the lowest compared to other models. Which were multi layered perceptron neural network

(MLP-NN), locally weighted regression (LWR), support vector regression (SVR) and online support vector regression (OLSVR) as shown in tables 1 and 2 below.

Detector Station (VDS)	Mean Absolute Percent Error (%)				
	MLP-NN	LWR	SVR	OLSVR	OLWSVR
*I5-N	5.2	5.3	4.9	5.1	4.4
**I5-N	7.7	6.3	6.1	5.4	5.0
SR101-N	8.0	5.5	5.2	4.8	4.1
I5-S	9.9	9.0	8.8	9.0	8.1
I10-W	6.7	6.5	6.1	5.1	5.6
I5-S	8.3	6.3	6.5	7.1	6.1
I880-S	5.2	5.0	4.8	4.6	4.2
Average	**7.3**	**6.3**	**6.1**	**5.9**	**5.3**

*POST MILEAGE: 12.75
**POST MILEAGE: 25.47

Table 1
Scenario 1

Detector Station (VDS)	Mean Absolute Percent Error (%)				
	MLP-NN	LWR	SVR	OLSVR	OLWSVR
I5-N	8.1	7.4	7.7	7.5	7.0
I580-W	14.0	12.2	10.0	10.5	8.8
I880-S	15.0	14.5	13.2	12.4	11.0
SR170-S	19.2	19.2	18.5	15.0	13.5
I5-S	10.5	10.9	10.7	10.4	9.9
SR57-N	13.6	13.4	12.9	12.8	12.5
SR91-E	22.8	22.9	22.4	23.4	20.5
Average	**14.8**	**14.4**	**13.6**	**13.1**	**11.9**

Table 2
Scenario 2

(Hu and Gao 2014) integrated particle swarm optimization algorithm with support vector regression to predict traffic. Particle swarm is an optimization algorithm has seven total steps which starts by calculating starting position of each of the particles in dataset. Then moves to check for best fitness of each particle. If the fitness of particle isn't at best, it fits it accordingly. Then again goes to step one by calculating fitness. Then checks for fitness again. In last two steps it first removes previous fitness values and updates it with the new one. And in the end finishes the algorithm. It is then integrated with the SVR. The dataset was taken from the second ring road in Beijing. It was based on six days total. Data was trained first on five days and tested on the last day. The results showed that SVR in addition with PSO predicts the results with lesser MAPE. As compared to multiple linear regression and Back propagation neural network.

(Jiang and Zhang 2022) Built a model as extension of the basic Kalman filer by combing it with second order Volterra (SOV-KF). Due to the nonlinear relationship in the traffic flow data The Kalman filer model traditionally used cannot deal with it. Second order Volterra can handle nonlinear data. Gaussian assumption is not fulfilled by the traffic flow data that's why the minimum least square criterion (MMSE) does perform well when it comes to non-Gaussian timeseries. Maximum Versoria criterion is introduced to tackle this challenge by introducing a new prediction model called SOV-MVKF. Results show that SOV-KF and SOV-MVKF models do a better job in predicting short term traffic.

(Xiao et. Al 2018) this research takes a different approach by focusing on estimating speed of the flow of traffic to predict future state of flow. Inductive loop detectors measure the route of the traffic and how many objects passed through the loops. It doesn't estimate speed of the flow. That's why to enhance the accuracy of future flow prediction based on the data from ILD's. The model developed here is a multiple-kernel (MKL-SVR). The algorithm is compared with three other commonly used time series models polynomial fitting, BP neural networks and baseline SVR. Results show that MKL-SVR gives higher accuracy compared with other models.

2.1.3 Deep Learning Models

Neural networks have changed the field of time series forecasting and improved the results in every field that time series forecasting. The methods being developed today are based upon decades of work. The idea is not new in any way. Since the traffic prediction work began in early 1970's. There were attempts to perfect to even more by adapting neural networks. That's why it's of extreme significance to look at the most important studies that have been done in this field over the period and include recent ones to know where we currently stand.

(Clark et. Al 1993) made one of the most important contributions in the field of traffic flow prediction by doing a comparative study between neural networks and ARIMA. Neural networks can be told from the input levels to describe the output levels. In the instrumented city of Leicester, the data is taken to be tested with 5 minutes interval with its link to the destination. Two things were to be investigated first one being the site of adjoining links and second one the chain links. Three statistical models are compared with neural networks. Results show that neural networks give higher accuracy.

(Demetsky and Smith 1994) an old study by them indicates that most of the studies early on focused on collecting data in intelligent highway systems. Even the way field was studied back then was different from now. Because of the complex nonlinear nature of the data. The need arose to look for more accurate ways to predict traffic flow. This introduced neural networks more and more into this area. Here three models based on historical average, data-based algorithms and time series model were compared go backpropagation neural network model. The backpropagation clearly showed superiority in handling the complex nature of nonlinear time series.

(Día 1999) took a different approach and predicted the travel time total from one point to the other. The data source used here is current loop detectors. Collected from the upstream and downstream of the station section. This will help in the prediction of how much time is needed to travel within these two points in future. This will help the drivers in getting alerts beforehand about the route they are taking. Dynamic estimates here can also help in alerting about incidents before reaching that point. Results show that neural networks can predict the travel time by 93% to 95%.

(Dougherty and Cobbett 1997) made a neural network to predict short term flow of traffic in urban areas. Backpropagation model was trained to predict the flow in Utrecht, Rotterdam, and Hague in Holland. The problem was that number of input parameters that go in the model were large. The neural networks that had all number inputs that were possible to be applied had the highest accuracy. But the large size of models didn't allow it to be implemented. To tackle this elasticity testing was introduced to reduce the function stepwise. The model's accuracy didn't improve significantly compared to other naive predictors.

(Día 2001) made another notable contribution by making an objected oriented neural network model for traffic flow forecasting. The model developed tested on the highway between two big cities Brisbane and Queensland. Feasibility of this method was checked by making a Time lag recurrent network (TLRN) which can predict speed up to 15 minutes ahead. The TLRN model can predict future flow with high accuracy but only up to 5 minutes. The other similar models developed were able to predict for 15 minutes ahead with accuracy significantly higher than conventional models. This gives this approach an upper hand and should be given preference to be applied.

(Lee 2007) made an important contribution by making a model that will predict traffic speed under weekday, time neighboring links speed This way instead of predicting flow in short term it will provide an alert to the system for the shortest time needed to reach the destination. For this advanced sensor were used to collect data. Due to sensor error or communication error some of the data was lost as well. Approach used here is a backpropagation neural network method then prediction is made under speed of neighboring links, time, and weekday as well.

Results show that method developed here reduces the error significantly compared to traditionally used.

(Park et. Al 2011) this study focused on travel time of the vehicle for multiple reason. It will help in traffic flow prediction, density of traffic and overall speed of the flow itself to determine trave time. This will then help in the optimization of vehicular operations, system for traffic controls and help driver by notifying them. Due to many factors that are dynamic in nature makes it difficult and a complicated task to do this accurately. A new approach called Neural Network Traffic modelling-speed prediction (NNTM-SP) algorithm is introduced for speed prediction. Combined with historical traffic data it will predict speed profile of vehicles and traffic information. Results show that accuracy of the model is high and reliable.

(Qiu 2011) this study focused on short term traffic speed prediction. The problem identified here is that many times the traffic flow prediction models lack satisfactory generalizability. To correct for this problem a solution is introduced Bayesian regularized neural network (BRNN). Regularization here is done by adding a weight decay function in the energy function that is a key component in structure of neural networks. One of the problems arises in deciding what parameters to add in weight decay function. BRNN is introduced to ease that challenge. Results of the study show that BRNN improves the generalizability of the model. And outperforms other traditional predictions models.

(Hosseini et .al 2012) another short-term traffic flow forecasting study which focuses on extracting mutual information (MI) from data and using neural networks to predict the flow. In modeling the models many times input used in the data is not up to the mark. And inputs used are irrelevant and not helpful in increasing the accuracy of prediction. Mutual information uses the most relevant inputs and helps in better forecasting of the data. Feature selection FS is an enhanced method based upon MI. This tested approach is called the MIFS algorithm. Results show that MIFS predicts the traffic flow with high accuracy.

(Zhao and Song 2015) came up with a model in their study which combines the two models of time series and classification. First is called graph convolutional network. And the other is called gated recurrent unit. Both were combined and resulted in temporal graph neural network (T-GCN). Wherever in nodes (data) there exists complex topological structures, GCN can be used to learn spatial dependence in them. Whereas GRU is used for the changes in data overtime. In other words when information between nodes which is carrying data is changing dynamically. The authors of this study predicted the traffic in urban road networks. Two datasets were chosen for this experiment one was from Shenzen and Los Angeles. The results showed that when compared to the baseline deep learning models like models like GCN and GRU. And two statistical methods like HA and ARIMA. T-GCN forecasted with higher accuracy and had lower RMSE.

(Tian 2015) developed an LSTM RNN model for predicting short term traffic flow. His can help in route guidance and efficient management of traffic. Before trying this model, several other algorithms and models were tried and tested none of them gave satisfactory accuracy. Most of the models require static input of historical data. This cannot determine time lags its own. LSTM RNN model here considers those problems and fixes it by dynamically determining optimal time lags for prediction. The modes used her for comparison are random walk, support vector machine, feed forward neural network and stacked encoder. The data used is real time PeMS from California, USA. Results show that LSTM RNN model outperforms all other models.

(Yu et. Al 2017) over the past decade the baseline models have improve many folds and the prediction accuracy has increased a lot. An important method to look here spatiotemporal recurrent convolutional networks (SRCN). This is a representation of network grid method. It can retain the fine scale structure of the network of transportation. The traffic in the entire network is converted into static images. Then it is used as input for the model. This model includes the advantages of both DCNN and LSTM. Data here used is from China with 278 links overall. Results show that SRCN's outperform other deep learning-based models both short term prediction and long-term prediction.

(Zhao et. Al 2017) developed an LSTM model to forecast short-term traffic. Goal was here to build a model that can help in alerting us with appropriate travel modes, travel time and departure times. This all is important and essential in traffic management. Deep learning approaches help us to consider most of the data that has been collected. Because they have more ability than statistical method used previously. The method here consists of two-dimensional network which is made up of many other memory units. Comparison is made with other forecasting methods. Results show this approach outperforms all others and predicts traffic with high accuracy.

(Yang et. Al 2016) made a deep learning model to forecast optimal flow of traffic. This will help in efficient management of traffic and better driving and safety for the drivers by reducing congestion. Stacked autoencoder Levenberg-Marquardt (SALM) model is proposed here for these purposes. SALM is designed from a Taguchi method which develops a structure that is optimal. It will then learn optimal features from the data layer by layer using a layer wise greedy algorithm. The data used is from MG motorway in Britain. It is then compared with three commonly used models. Results show that proposed approach outperform the other models.

(Gu et. Al 2019) made a model called spatio temporal graph attention base CNN. The traffic data when collected on a large scale, which in today's technology makes it totally possible. Follows complex nonlinear form. Which makes it difficult to predict traffic with higher accuracy that is more reliable to be deployed in real times. Most of the models used cannot overcome the difficulty of predicting data with spatiotemporal correlations. ASTGCN makes it easier to pass those challenges. This approach has three independent components weekly periodic, daily periodic, and recent. Model is compared with other state of the art models for prediction. It is used on PeMS dataset. Results show that proposed approach gives higher accuracy compared to other models.

(Wu el. Al 2018) made a graph-based attention LSTM (GAT-LSTM) for traffic flow prediction. It becomes difficult in traffic flow prediction to predict the traffic on the roads when there are to man links meaning other roads joining in. Graph based neural network in particular LSTM model can help fix that problem. This approach uses and end to end trainable encoder forecaster model to fix the problem of multi-link traffic flow forecasting. Results of this approach shows that it improves the forecasting accuracy significantly compared to other state-of-the-art baseline models.

(Yu et al. 2017) developed a framework based on deep CNN for spatiotemporal traffic flow forecasting (STGCN). Because of the low accuracy of traditionally used models. This model is trying to fix that and increase accuracy for better guidance in traffic management. The problem is formulated on graphs and applying regular recurrent units. Fewer parameters will be used for faster prediction of future flow. The structure of the model will be completely convolutional. Results show that STGCN captures all the spatiotemporal correlation by successfully modelling multi-scale network. Compared to other prediction models used the accuracy of this approach increases significantly.

(Albertango and Hassan 2018) used deep learning to solve traffic congestion problem in the longer run by timely predicting short term traffic. Which will result is better management overall. The better idea to collect data is to use limited number of sensors to capture traffic volume information. This way by combining it with AI techniques. The cost on ILD's and other expensive ways can be saved. Results of the model show that it predicts the short-term traffic with good accuracy. And do so even without a lot of training on prior flow.

(Cai et al. 2016) develops and extension and an improved version of KNN. A statistical method widely used in time series prediction. The improved model is combined with other methods. This makes the model good enough to capture the spatiotemporal correlation and will be able to predict multi step ahead. The distance among the roads from each other is defined by static and dynamic data which is collected from real road networks. State of the traffic is defined by a spatiotemporal state matrix. Nearest neighbors are selected according to the Gaussian weighted Euclidean distance. Compared to four other models used to predict traffic. Results show that improved version of KNN gives higher accuracy for short term traffic forecasting.

(Cui et. Al 2018) made a model which predicts bidirectional and unidirectional traffic flow data (SBU-LSTM). The depth and full potential of LSTM has been rarely applied. This approach focuses on exploiting model architecture, scale of spatial area and spatiotemporal data's predictive power. This model accounts for both forward and backward dependencies present in the data. This is the first time a BDLSM layer is used in the prediction of traffic flow. Model can also take care of the missing values by mechanism of masking. Results show that when compared other models. SBU-LSTM outperforms them significantly. It manages to predict the traffic flow with superior accuracy.

(Han et. Al 2019) developed a model for highway traffic flow forecasting. Spatiotemporal features in traffic data hold significance influence over forecasting. Internal relationship among the roads that are adjacent spatial information can affect that forecasting. Periodic flow of that can also affect temporal features. This model will learn features from time and space dimensions. CNN are used to extract the spatial part meanwhile LSTM are used for the temporal features. The connected structure of LSTM and CNN are powerful in traffic flow prediction. Data used is from Shanghai, China collected from 591 sensors. Results show that this deep learning model surpasses state-of-the-art models in accuracy.

(Lv et al. 2015) made a deep learning model to predict traffic flow forecasting with big data. Accurate measurement of flow is important part for successful ITS. It is easier than it used to be in the early years of the field to collect data. In fact, now we are in the phase of field called 'Big Data'. Many of the existing prediction models make shallow prediction not applicable in real time and not good enough to handle large amounts of data. This model is a stacked encoder model which learn generic traffic flow features from the given data. Results show that the approach proposed here gives higher accuracy compared to other baseline models.

(Parnami et. Al 2018) made a deep learning-based model for urban analytics platform (UAP). Cites have become increasingly wired and interconnected with each other. Carrying a lot of data within. This gives rise to a lot of opportunities for advancement in the field of ITS. Which will in turn help build better frameworks for future urban centers. This approach tries to contribute into that. UAP is an end-to-end system for processing and accounting for all that data sources. Neural networks based deep learning models will be used in this system. Results show that this approach improves the traffic prediction and can be used in alerting about the weather and other pattens that affect the flow.

(Polson and Sokolov 2017) developed a deep learning model for short term traffic flow prediction. Most notable part of this model is a sequence of tanh layers and the L1 regularization. Because of congestion, breakdown and sudden fast flow of traffic sometimes slow pauses cause non linearities in the data collected. This pushes us to focus on developing models that overcome challenges of nonlinearities. Data collected for testing this model is form I-55 highway for two specific events. Results show that proposed approach provides high accuracy in prediction of flow in both events.

(Shafqat et. Al 2019) developed a recurrent neural networks-based model for short term traffic flow prediction. The approach here is to collect traffic data from different sources. But this won't include only data from different sources but will focus on collecting data from different conditions as well. For instance, it is possible that at one point due to tight space traffic is congested most of the time at some peak hour. But on a different road the built of the road is on a dangerous path which causes accidents and then congestions due to that. Results show that proposed model predicts traffic with accurate flow at any given time interval.

(Yu et. Al 2018) developed a hybrid deep learning based model for traffic flow prediction (DNN-BTF). The currently used deep learning models give better results than old statistical and machine learning models. They are yet to be used to their full capacity to include spatiotemporal characteristics of traffic flow. DNN-BTF exploits the weekly/daily and spatial temporal parts of the flow to the fullest. The CNN is used to account for spatial features and RNN is used to account for temporal features. The dataset used for testing is the PeMS dataset. Results show that this DNN-BTF outperforms stat-of-the-art models used for same prediction.

(Yang et. Al 2018) made short term traffic prediction model based on spatiotemporal features of critical road sections. Traffic prediction with missing data or data that is corrupter has gotten attention from researchers recently. A specific road network on a location has connection with adjacent roads. CRS-ConvLSTM NN model is going to predict the traffic evolution of all those networks. Spatiotemporal correlation algorithm is used to identify road sections that impact subnetworks the most. Data used for the study is from Beijing, China. Results show that CRS-ConvoLSTM model outperforms other deep learning models and gives higher accuracy.

(Zhang and Kabuka 2018) made a deep learning model based on Gate Recurrent Unit (GRU) which includes weather data. It is important to test models inclusive of weather data. Due to the simple reason on simulation software's one can make as many hypothetical situations run models and check results. The real weather data collected from sources. Combined with real time traffic data. Always outperforms the simulations. This is the first-time weather data has been combined with real time traffic data to be precited by GRU method. Results show that prediction accuracy improves when weather data is combined with traffic data.

(Chai and Wang 2018) implemented a multi graph convolutional neural network (MGCN) for bike flow prediction. The purpose was of author was predict the bike flow at station level. The edges in the graph represented a relation the nodes. While the nodes were the stations which were being predicted. First multiple graphs were drawn and the different relationship between each of them was analyzed. After that all the graphs were fused into one. Once the graphs were fused then one by one layers were applied to them.

In the end the prediction accuracy of bike flow at station level increased. The test case was run on two big cities New York and Chicago bike station system. In New York's case prediction error decreased by 25.1% and by 17.0% in Chicago. The authors also gave a confidence interval after running simulations unlike another research we see on this topic. This can help managers make better decisions. Although this study was done on a bike station dataset. The same logics and model can be applied to any real time, real-world benchmark traffic datasets like mentioned in the previous studies.

(Zhang et. Al 2019) developed a model for flow prediction in spatiotemporal networks based on multi-tasking deep learning models. Nodes that have in and out traffic and then that flow transitions from one node to another in spatiotemporal networks plays a critical role in transportation systems. This becomes a difficult task to predict accurately. Due to factors like spatial correlation, temporal correlation among two different time intervals and weather, incidents, or other factors like these. This approach looks to overcome these challenges. Data used for testing is from New York and Beijing taxicab data. Results show that model outperforms 11 baseline models.

(Zhang et. Al 2019) made model that predicts short term traffic based on spatiotemporal analysis and deep CNN learning. Most of the bassline models fail to make full use of spatio temporal features present in the traffic data. In this model the optimal amount of input for time lags and amount of spatial data are determined by spatiotemporal features selection algorithm (STFSA). The CNN part of the model then is trained on these features. Real time traffic data is used for testing. Results show that after comparison proposed model outperforms commonly used baseline models in terms of accuracy.

(Zheng et. Al 2019) developed a deep embedded learning (DELA) approach for traffic flow prediction. This method will be inclusive of weather condition and other conditions that could affect traffic. It includes three components first one being CNN, second LSTM and third which makes this stands out from different model the embedded component. CNN learns form 2-

D traffic structure and LSTM maintain long term historic data. Embedded can capture categorical features and help in identifying correlation information. Data used is real time and realistic. Results after comprehensive experiments show that DELA outperforms existing models in prediction.

(Zou et. Al 2018) made a model to predict city wide traffic via LSTM networks. Because of the uptick in vehicle purchases in recent times. And more and more people preferring to use their own means of transportation. This causes trouble for the environment by leaving a lot of gas energy in the atmosphere and other severe traffic related issues. Making monitoring of traffic a critical issue for long term better and reliable source to predict and manage it efficiently. Two benchmark datasets were used to predict the traffic and test the model. Results show that LSTM give high accuracy making it reliable for real time deployment.

Another important contribution in the field of spatio-temporal forecasting was done by (Medrano and Aznarte 2020). They constructed a model called Convo-Recurrent Attention Network (CRANN). It is the combination of both CNN and RNN. RNN due to its recursive nature and CNN due to it being able to classify images and spatial changes over time with higher precision. The temporal module in their model was made up of two layers of long short-term memory (LSTM). The first layer takes input from the data fed. After calculation it passes out a hidden state. Which then is coupled with a 'context vector'. And then second layer is used to make prediction for next time step. For spatial module was developed by weight of input, attention function and dot product, vector score which was taken from CNN which gave a score x_{conn} and weight of input.

Both spatial and temporal combined made a model which was then ready for prediction. Forecasting was done on dataset from Madrid, Spain. Which was 24 hours long. To get results with higher accuracy. Apart from traffic authors also added weather data of Madrid. The results showed that CRANN as outperformed state of the art modes like CNN, LSTM, CNN+LSTM and seq2seq models.

(Menkinen et. Al 2022) compares models to predict short term traffic flow. And see which one performs better in mid-sized European cities. The models compared here are the ARIMA family models, Linear regression, KNN's and XG Boost models. The prediction horizon or in other words interval time used from the dataset is 5, 10, and 15 mins. Dataset itself is open data from Finnish capital Helsinki. Sudden speed drop in traffic usually implying traffic congestion and traffic jams. Is paid attention towards carefully. Because it tells a great deal about flow of traffic overtime. Results show that XG Boost model outperforms all other models over all the intervals. But classification models like decision trees tend to do better when it comes to traffic jams.

(Zhang and Liu 2021). They developed a model that forecasted traffic with spatio-temporal multi headed attention networks. This model uses graph attention networks for capturing the spatial dependence in the data and then after scaling the dot product it positions the encoder like transformers to extract the temporal dependence from given data. They used their models on two benchmark data sets and compared it with the previously used statistical models (SM), machine learning (ML), models and deep learning (DL) models. Results showed that proposed model gave higher accuracy than all the other SM, ML or DL models.

(Ding 2021) made an intelligent algorithm for the prediction of short-term traffic. It will be helpful in better traffic management, better decision making, and is one of the basic parts in intelligent transportation systems. The traditional mathematical models are not good enough to use for these systems. Backpropagation method and redial basis neural networks are studied here. The on-spot investigation of traffic flow is studied through them. Real time survey data is used for evaluation of these algorithms. Results show that radial basis function (RBF) does a better job at prediction.

(Pavlyuk 2022) compared existing models used to learn spatiotemporal relationships from traffic data for the better prediction of flow. Many algorithms have been used for the purpose of studying space and time like cross-correlation, graphical lasso, mutual information, and random forest. Two issues are discussed one is the application of ensemble learning techniques. Second is the recommendation of strategies that investigates their dynamics and provides insights into the responsive learning of spatiotemporal relationships. Results show that ensemble learning techniques do better than other techniques. And the recommended strategies are preferred over others.

(Selwi et. Al 2022) made models to predict traffic flow. First the model was without including weather data and second was with the inclusion of traffic data. It is important in the longer run to keep this question into perspective that weather can affect the flow of traffic or not. And ask this question to what degree will it affect the model's prediction. The irregularities caused by occasional disruptions in the flow of traffic. Could add up after year or a quarter worth of data and give wrong information. Results of comparison here showed that with inclusion of weather data models predict flow with higher accuracy.

(Wang et. Al 2022) developed a feature map and deep learning model for traffic state recognition. It is important to know the traffic state to have bigger picture for efficient management of traffic. Traffic flow prediction alone makes it difficult to show us that. Traditional approaches for traffic state identification use only basic feature vectors and ML models. Which doesn't benefit fully from deep learning methods. In training phase Deep cluster algorithms is used to study the vehicle trajectory meanwhile in online phase CoAtNet algorithm is used. This is then compared with existing deep learning models. Results show that proposed approach give higher accuracy.

(Xu et. Al 2021) developed two model autoregressive fractional integrated moving average (ARFIMA) and compared it to ARIMA. This was going to useful to predict the linear component in the traffic data for future prediction. The second mode developed is Nonlinear autoregressive (NAR). This is for the nonlinear component of the data. And is going to be compared to singular models. PeMS open access dataset from California is used for testing the models. Results show that ARFIMA does better than ARIMA and NAR does better than singular models.

(Nishar and Kumar 2022) developed a model that is multi-branch and predicts traffic based on temporal speed. There have been various studies up until this point that predict traffic based on the combination of spatial and temporal models. Traffic flow theory tells us that flow of traffic and its speed are related. There are two branches here in this model keeping this relation in view. Using past flow data first branch predicts traffic based on LSTM. Second branch predicts volume using Gaussian process regression (GPR). Results show that models perform well and give high accuracy.

(Gu et. Al 2023) made a dilated convolutional network (DCN) with peak sensitive loss. DL models by enlarge are effective and reliable in predicting traffic flow. But they are ineffective at wide range spatial correlations. This model will add a cost sensitive loss function to account for the loss of the results. This method can improve the performance because mean square loss and square of mean absolute percentage are difference types of costs employed metrics. Temporal convolutional network (TCN) is also constructed. Results show that both models are effective in real time deployment.

(Tang et. Al 2023) made a model to study conjoining speed cycle patterns and predict traffic speed using neural networks. Usually, congestion is caused by regular activities during the flow of traffic. Neighbor subset deep neural networks (NSDNN) are used for spatiotemporal forecasting. Subset selection method is conjoined with DNN to extract useful insights from the roads connecting to the main one which flow if being predicted ahead. Results show that compared to the models like ARIMA, KNN, LSTM and NLSTM NSDNN gives higher accuracy.

(He et. Al 2023) developed a GNN based traffic prediction framework with spatiotemporal granger causality graph (STGC). It is of extreme importance to accurately predict temporal dynamics of flow. The existing models are local and spatial dependence. Which cannot predict long term traffic because it cannot transmit global-dynamic traffic information (GDTi). The challenge here is that for individual vehicles it's difficult to predict GDTi. The approach here is to use transmitting causal relationship (TCR) to study underlying traffic flow. And use STGC to express TCR which will then model global and dynamic spatial dependence. Results show that STGC dos a better job than other models and gives higher long term flow accuracy.

(Liu et. Al 2023) developed a GraphSage based dynamic spatiotemporal graph CNN (DST-GraphSage) for traffic flow prediction. The complexity of spatiotemporal dependencies makes it difficult to predict flow accurately. Most of the DL models that are developed relay on extracting spatial and temporal dependencies separately. DST-GraphSage can extract dynamic spatiotemporal dependencies at the same time. Making the model more efficient. And additional layer of dilated causal convolutional to extract temporal part of the data has been added. Results on five real time traffic datasets show that proposed model in effective and gives high accuracy.

(Ting et. Al 2021) did an extensive study in which they compared model developed over time for the traffic flow prediction. Models included here are time series models which include statistical models as well as machine learning models. Linear regression models are also included along with more recently developed deep learning models. The models are all baselines. And are tested on greater Toronto area simulations for traffic prediction. Results show that deep learning traffic models including Graph CNNs are the most effective for traffic flow prediction. However, ensemble methods can give even higher accuracy but take a lot of time to be built perfectly which can give higher accuracy than DL models.

(Wang et. Al 2022) also did a detailed comparison of ML and DL models based on baseline models and mixed models. The models included are LR, SVR, RF, DT, LSTM and GRU. Real time data from England's highway is used. Results show that in both cases baseline models and mixed models approach based on velocity distribution. GRU outperforms all other models and gives the lowest MAE.

2.1.4 Discussion and Analysis:

The field of deep learning is evolving fast. In past ten years alone there has been significant improvements in the way predictive analysis is done whether it is prediction or classification purposes or it's the time series prediction. For our purpose of predicting the traffic flow better and in the end reducing the congestion overall there are a lot of choices of different algorithms and models that one can pick from.

There are two ways in which now data models are prepared for predictive purposes the first one is spatial and second being temporal. The spatial prediction includes first analyzing the videos and images feed of a certain traffic point where traffic is to be forecasted for congestion reduction. And then based off that the model is developed to forecast. Meanwhile temporal predictive model includes only the time series of a collected data or real time streamed data from a traffic point where traffic is to be forecasted. Temporal models are then fed the data of the timeseries for forecasting the flow.

When we look at big city centers now around the world for instance Los Angeles, Beijing, Istanbul, New York, and Jakarta experts have been able to find both these types of data. Which has given rise to new types of models that predict traffic based off spatiotemporal data. This means that both types of data are now included in these models. But most of the cities around the world due to cost constraints haven't been able to induct advanced traffic cameras or even real time data collecting devices. ILD's are now common in North America, Europe and developed countries of Asia or rapidly developing countries.

Oh et. Al 2002 showed that conventional loop detectors are not the most accurate tool for recording the parameters like speed of vehicle, occupancy on road (for traffic congestion purposes) and volume of car. The accuracy it gave was around 90% to 95% every time. ILD's single or double lane both outperformed the conventional ones. For the spatial data purposes there is no cheap way around the cities will have to invest in the live traffic capturing high-quality, high-resolution cameras. This will not only help in spatial analysis of traffic prediction in the shorter run. In the longer run it could also help a great deal in collecting data for making algorithms specifically for congestion purposes.

Table 3 below shows the detailed comparison of the algorithms and models used for traffic prediction purposes. This isn't based on any one type of model. It shows the different studies over the period done based on spatial, temporal, and hybrid, meaning the combination of both. It might seem obvious to anyone that hybrid would give the best results but is that true? The answer to this can be more complicated than one might assume.

The goal in the end of all this effort is to keep the traffic flowing and the reason for that is making sure that congestion is avoided. Although nowadays due to budgetary constraints and other issues the researchers have been focusing on straight up developing the congestion algorithms which then advice on how to design roads, develop better city centers so driving becomes more friendly and traffic doesn’t get congested. One of the ways to do so is adding more lanes to existing infrastructure.

Handy (2015) through their study have shown that if the capacity of existing road is increased meaning another lane is added to roads network. This is likely to increase the total number of vehicles travelling on that road by 3% to 6% in shorter run. Meanwhile the in longer run it is estimated to be increased by 6% to 10%. If tackled a situation like this, it is better to develop a rail transit system or strong public bus transport system. The models developed for traffic prediction will not be able to give results that will help in better flow prediction to reduce congestion.

That is why models that show spatial and temporal data below in table 3 cannot be used just because the relevant traffic experts might think that they are better than the others because they allow both types of data to be processed. The flow forecasted in the city center is different from the flow forecasted on a highway. The end goal is certainly the same. When making models it is important to consider finance, location, and purpose in shorter and longer run both. One study mentioned above has already shown that simple statistical model like ARIMA performed better to predict traffic in shorter run. But in the longer run it failed to keep up with deep learning models.

These are some of the reasons that a wholistic approach should be taken for completing a complex task such as continuously predicting traffic flow. The research in this field is still ongoing before we can have a definitive answer on which algorithm is better than the other one and in what scenario. For now, deep learning model for the most part outperform the simple machine learning and statistical methods. Even in deep learning models there is contrast and matter of empirical analysis on which model should be relayed upon.

Research is still going on before we can have "one size fit all" kind of model for our better management of traffic. And even afterwards we must not forget that intelligent traffic management is one component in the overall intelligent transportation management. Every component will have its limitations individually but when combined. It can potentially help us manage transportation in our rapidly urbanizing world intelligently.

Table 3: Summary and comparison of SM, ML, and DL models for Traffic Flow Forecasting.

Model Description	References	Spatial	Temporal	Data Source
ARIMA (SM)	(Ahmed and Cook 1979)		*	Historical Data
Historical Average (SM)	(Stephanedes 1981)		*	Historical Data
ARIMA (SM)	(Stephanedes and Okutani 1984)		*	Historical Data
Comparison NN Vs ARIMA (DL)	(Clark et. Al 1993)		*	Historical Data
Comparison BP Vs HA Vs DBA Vs TS (DL)	(Demetsky and Smith 1994)		*	Historical Data
Comparison KARIMA Vs ARIMA (SM)	(De Voort et. Al 1996)		*	Historical Data
Nonparametric Regression (ML)	(Smith and Demestky 1996)		*	Historical Data
BP (DL)	(Dougherty and Cobbett 1997)		*	Historical Data
Comparison Subset ARIMA vs ARIMA (SM)	(Lee and Fmbro 1999)		*	Historical Data
NN (DL)	(Día 1999)		*	Historical Data
Comparison KNN Vs SARIMA (ML)	(Lee et. Al 2000)		*	Historical Data
Comparison ARIMAX Vs ARIMA (SM)	(William 2001)		*	Historical Data

TLRN (DL)	(Día 2001)		*	Historical Data
Pattern Matching (ML)	(Clark 2003)		*	Historical Data
SVR (ML)	(Wu et. Al 2004)		*	Real Time Data
Knn extension (ML)	(Kim et. Al 2005)		*	Historical Data
Comparison SARIMA Vs ARIMA (SM)	(Gosh and Basu 2005)		*	Historical Data
Comparison SARIMA Vs ARIMA (SM)	(William and Hoel 2005)		*	Historical Data
BPNN (DL)	(Lee 2007)		*	Historical Data
Comparison Kalman Filter vs Wavelet KF (SM)	(Xie et. Al 2007)		*	Historical Data
Kalman Filtering (SM)	(Wang et. Al 2008)		*	Historical Data
AOSVR (ML)	(Zheng el. Al 2008)		*	Real Time Data
Multivariate structured (SM)	(Gosh and Basu 2009)		*	Historical Data
Comparison OL-SVR Vs GML Vs ES Vs Ann (ML)	(Castro-Neto et. Al 2009)		*	Historical Data
Comparison SVARCACO Vs SARIMA (ML)	(Chiang Hong et. Al 2011)		*	Real Time Data
NNTM-SP (DL)	(Park et. Al 2011)		*	Historical Data
BRNN (DL)	(Qiu 2011)		*	Historical Data
Comparison WPRA Vs PRA (ML)	(Li et. Al 2012)		*	Historical Data
Comparison K-LWR Vs Knn (ML)	(Li et. Al 2012)		*	Real Time Data
MIFS (DL)	(Hosseini et .al 2012)		*	Historical Data
SVR (ML)	(Asif et. Al 2013)	*	*	Historical Data
OLWSVR (ML)	(Jeong and Byon 2013)		*	Real Time Data
PSOSVR (ML)	(Hu and Gao 2014)		*	Real Time Data
Comparison SVR Vs LR-SVR (ML)	(Ahnn et. Al 2015)	*	*	Real Time Data
LSTM-RNN (DL)	(Tian 2015)		*	Real Time Data
Comparison ARIMA Vs SARIMA (SM)	(Kumar and Vanajakhsi 2015)		*	Historical Data
HMMKMM (ML)	(Hong et. Al 2015)		*	Real Time Data
Stacked Encoder (DL)	(Lv et al. 2015)		*	Historical Data

Comparison T-GCN Vs CNN Vs GRU	(Zhao and Song 2015)		*	Historical Data
Interval Type 2 Fuzzy Sets Theory (SM)	(Li 2016)		*	Historical Data
Deep KNN (DL)	(Cai et al. 2016)	*	*	Historical Data
SALM (DL)	(Yang et. Al 2016)		*	Historical Data
TRU-VAR Vs ARIMA (SM)	(Schimbinschi et. Al 2017)	*	*	Real Time Data
SRCN (DL)	(Yu et. Al 2017)	*	*	Real Time Data
LSTM (DL)	(Zhao et. Al 2017)		*	Historical Data
STGCN (DL)	(Yu et al. 2017)		*	Historical Data
DL-L1 (DL)	(Polson and Sokolov 2017)		*	Historical Data
Comparison MKL-SVR Vs BP Vs PF SVR (ML)	(Xiao et. Al 2018)		*	Historical Data
LSTM (GAT-LSTM)	(Wu el. Al 2018)		*	Historical Data
UAP (DL)	(Parnami et. Al 2018)		*	Real Time Data
DNN-BTF (DL)	(Yu et. Al 2018)	*	*	Real Time Data
MGCN (DL)	(Chai and Wang 2018)		*	Historical Data
SBU-LSTM (DL)	(Cui et. Al 2018)	*	*	Real Time Data
GRU (DL)	(Zhang and Kabuka 2018)		*	Historical Data
CRSConvLSTM (DL)	(Yang et. Al 2018)	*	*	Real Time Data
LSTM (DL)	(Zou et. Al 2018)		*	Historical Data
AMSVM (ML)	(Feng et. Al 2019)		*	Real Time Data
ASTGCN (DL)	(Gu et. Al 2019)	*	*	Real Time Data
MTDL (DL)	(Zhang et. Al 2019)	*	*	Historical Data
LSTM-CNN	(Han et. Al 2019)	*	*	Real Time Data
RNN (DL)	(Shafqat et. Al 2019)		*	Historical Data
CNN-STFSA (DL)	(Zhang et. Al 2019)	*	*	Real Time Data
DELA (DL)	(Zheng et. Al 2019)		*	Real Time Data
Hybrid Markov (ML)	(Yao et. Al 2020)		*	Real Time Data
CRANN-LSTM (DL)	(Medrano and Aznarte 2020)	*	*	Real Time Data
Comparison SFRF Vs GBDT Vs HA Vs	(Tian et al. 2021)	*	*	Historical Data

RF (ML)				
RBF (DL)	(Ding 2021)		*	Real Time Data
Graph Attention Network (DL)	(Zhang and Liu 2021)	*	*	Historical Data
Comparison Graph CNN and DL Vs Other models (DL)	(Ting et. Al 2021)		*	Historical Data
Comparison ARFIMA and NAR Vs ARIMA and other Models (DL)	(Xu et. Al 2021)		*	Real Time Data
Timeseries Decomposition (ML)	(Huang et. Al 2022)		*	Historical data
SOV-KF and SOV-MVKF (ML)	(Jiang and Zhang 2022)		*	Historical Data
Comparison ARIMA Vs LR Vs Knn Vs XG Boost (DL)	(Menkinen et. Al 2022)		*	Historical Data
Comparison Ensemble Learning Vs Cross Corr. Vs LR Vs Lasso Reg. (DL)	(Pavlyuk 2022)	*	*	Historical Data
Comparison With Vs Without Weather Data (DL)	(Selwi et. Al 2022)		*	Historical Data
Comparison Deep Cluster, CoATNeT Vs Deep Learning (DL)	(Wang et. Al 2022)		*	Historical Data
LSTM-GPR (DL)	(Nishar and Kumar 2022)		*	Historical Data
Comparison LR, SVR, RF, DT, LSTM, and GRU (DL)	(Wang et. Al 2022)		*	Real Time Data
DCN-TCN (DL)	(Gu et. Al 2023)	*	*	Real Time Data
Comparison NSDNN Vs Other models (DL)	(Tang et. Al 2023)	*	*	Historical Data
STGC (DL)	(He et. Al 2023)	*	*	Real Time Data
DST-GraphSage (DL)	(Liu et. Al 2023)	*	*	Real Time Data

2.2 Image Classification

An important component in the feedback and analysis for effective traffic management is image processing and classification. It is important to look at the recent literature in these approaches. The methods for image classification and models made can be integrated in the hardware and infrastructure for Intelligent traffic management. Because technology is simply going to enhance the abilities of authorities and make it more efficient for them to do their work. A study done by (Hooda and Yadav 2016) shows that how traffic lights can be dynamically timed according to the congestion of traffic. This way traffic will keep flowing as the AI methods inside the system will be able to calculate the traffic density. After which without any traffic authority's interference the signals can be helping in keeping the traffic flow going. Most of the signals have traffic cameras tied to them. Even if in many places there are no loop detectors or other advanced technological gadgets. This is where this proposed technique is effective it can be integrated into existing infrastructure like CCTV and advanced traffic control (ATC).

(Papageorgiou 1998) and colleagues presented a general framework for object detection. This is seen as one of the first studies that presented such an extensive framework in the field of object detection. Method presented in this study is applied on the static images with scenes that are clustered. The technique for detection presented here is applied using walvet representation of object classes that are derived from the statistical analysis that are done on class instances. First the object classes are learned by the model as an overcomplete dictionary in terms of walvet basis function. This is a subset of the dictionary.

After this the function is used in a complete form as an input for the Support Vector Machine (SVM) classifier. Doing this overcomes the challenge of in class variability. And the false detection rate declines in unconstrained environment. This method is tested by authors in two systems. In which the content of both differs from each other significantly. In the first part of system face is detected. In the second part the domain is detected which means the patterns, colors, and texture. This system learns completely from training and does not rely on any previous models or training experience. In this method another motion-based extension model was added to further enhance the algorithm. The results of this method showed promising results that can be generalized.

The authors of this study used an Ada boost algorithm. This algorithm was fed images of various kinds of accidents, congestion, bad or good. Which were overall classified into two categories 1800 negative and 1200 positive. SUMO simulator for 4000 steps was used. The results showed that this algorithm if integrated with the existing traffic infrastructure can be efficient for traffic management in many ways. It can help in congestion control, internet of all things in further data gathering, emergency vehicle preemption etc. Not only that it also helps in helping save fuel and energy. Which puts less load on the ecology by keeping the traffic going.

One of the main points of focus in the research for ITMS has been reducing congestion for emergency vehicles. This doesn't usually become a problem. But in many cases the traffic simply manages itself accordingly there are delays and jams which hurt the patients or other emergency services. A model proposed by (Sundar and Hebbar 2015) shows how radio frequency identification (RFID) can be used. RFID can be placed at strategics locations throughout the city on main routes. All the big urban centers around the world always have a few key main roads that connect towards all the other main directions of the city. This type of system can be placed in those places to help the emergency vehicles and even stoles vehicles. According to the model proposed the RFID can read the RFID tags on the individual vehicles then it can store information on how much traffic has passed through this road. This will help in reducing network congestion. And it will automatically reduce the need for human traffic police intervention.

Not only that it can read the RFID tag of each vehicle and see which ta belongs to the stolen vehicle. This way it can help in detection of stolen vehicles as well. In addition, for the emergency vehicles the advantage of using this type of system is that it can directly contact the traffic room if there is a blockade or congestion ahead. The authors used ‘ZigBee modules on CC2500 and PIC16F877A system-on-chip for wireless communications between the ambulance and traffic controller’.

A key advancement in ITS has been in recent year by the introduction of Vehicular ad hoc networks. This is a concept straight for the future smart cities. It enables exchange of information not only between drivers due to interconnection of their vehicles. But also, the vehicles connected due IoT provide information to the vehicles. Although it is a completely different area of study. In ITMS it can be applied the way (Khekare and Sakhare 2013) as well. The authors put forth a VANET based framework for ITS in smart cities. In which create an Ad hoc Demand Vector (AODV) which send messages after every 2 seconds which then updates the statistics and changes the data accordingly. Under this scheme all the transportation on the road is connected on the road with each other by IoT. Which is why for no its applications are limited in the modern urban centers. Because all the vehicles do not have the necessary tools yet to join this type of system. The proposed framework is nevertheless relevant and helpful in the future of VANET. Where all the traffic lights and transportation on the road is interconnected by IoT. (Senthilnayaki 2022) proposed a model using inception V2 architecture. In this model a faster R-CNN is used in to predict traffic sign and then CNN is added. Blurring, blockage, or sharp light could make it difficult for the system in which these models are deployed to predict the traffic sign. The framework proposed by the authors can be used in variety of different fields like driving assistants and autonomous vehicles. The R-CNN first selectively searches to make an ‘area proposition boundary’ after which the CNN is taken out. After which the extracted features are fed to a model which predicts positions of objects.

After applying model on the benchmark dataset of German Traffic Sign Detection. Which consisted of 900 images that was for training data 100 for testing. The authors found out that ‘a valid match between the response field, reference window, and traffic sign is required.’ If the fields of response, windows of reference are reduced, and feature maps are increased. Then the faster R-CNN methods can be more useful and accurate than traditional methods.

(Altundogan and Karakose 2019) presented a physical feature determiner model for traffic stakeholder based on deep neural image classification by image processing. This model classifies traffic stakeholders in different categories like pedestrians, trucks, automobiles, busses, and other sources of transportation. This model can be integrated in the traffic lights in other vehicles or any such assets which the relevant authorities want to use as a source for identification. The proposed method first takes the traffic video provided through cameras in place. Once that’s done then it extracts contours from the video. After that the application starts by deep image classifier which then classifies these contours. Apart from this the model can also be applied to get vehicle dimensions as well. The benefits of getting dimensions are that it will make it easier in the smart cities and their researchers, people who are building and managing the city to get information on the size, type, and number of different types of vehicles. The dimensional features are based on the contours size meanwhile the color determined based on HSV features. Figure 2 shows the methodology by (Altundogan and Karakose 2019).

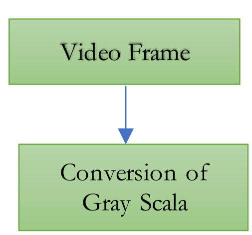
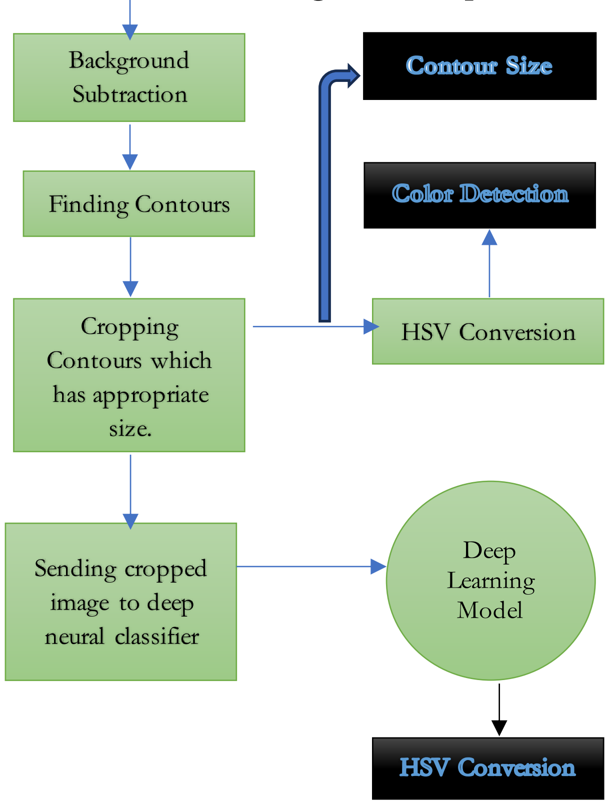
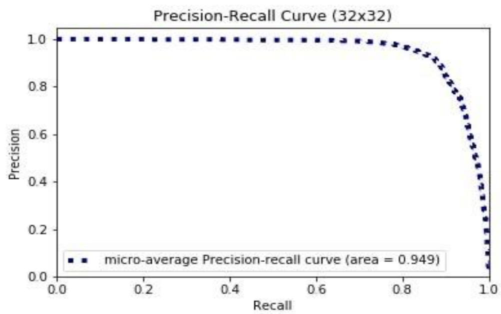


Figure 2. Deep Neural Image Classification



(Schuszter 2017) did a comparative study of machine learning methods for traffic sign recognition. All the details of key methods were presented. First approach used here is the Support vector machine (SVM) which used HOG descriptors. This method is deployed for the model to learn the different boundaries between the classes of traffic signs. Which show better results where the resolution is higher. The second model deployed is deep Convolutional Neural Networks (CNN). These neurons much like human eyes try to learn different features in any given image. The third method deployed here for comparison is Deep Residual Network (DRN) it is the most recent one as compared to previous two.

There were two different experiments done to extract HOG descriptive features. One was image size 32x32 and the second was image size 64x64. The larger image gave better results as shown in **Fig 3 and 4.** For accuracy score micro average way was used. Which aggregated all the curves into one single. For easier visual understanding.



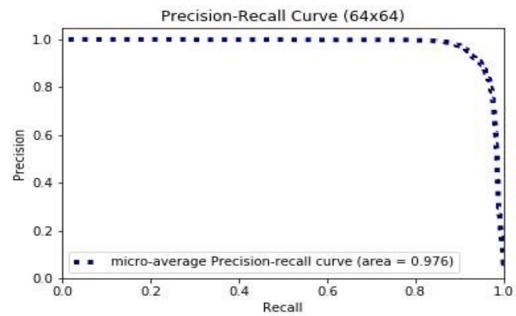


Fig 3. ROC curve 32x32 size **Fig 4. ROC curve 64x64 size**

Accuracy on test dataset was 70% for 32x32 image size and 86% for 64x64 image size. CNN test dataset accuracy score was 95.83%, ResNet18 test dataset accuracy score was 93.69% and ResNet50 dataset accuracy score was 89.92%. From the given score one can see that CNN gave the best results for traffic sign recognition.

(Dadi 2016) and colleagues presented a novel face recognition algorithm. Just like the previous study they also used Histogram of Oriented Gradient (HOG) features form both training data images and test data images. That were then fed to SVM classifier model. The algorithm is then compared with another common face recognition algorithm called eigen feature based face recognition. The proposed algorithm by the authors was used on 8 datasets. Principal Component Analysis (PCA) a data dimensionality reduction method was also used in this process. The dataset used for experimentation was from ORL AT&T database. The data was divided into overall faces of 40 people.

After applying the algorithm to the dataset, three color metrics were chosen for visual representation. If output is green color, then PCA is showing wrong faces as outputs, and which was corrected by using SVM-HOG. If the output shows orange as a result. This would mean that both the algorithm PCA and SVM-HOG are wrong. And if the color is red by the output, then it would be interpreted as PCA being right is classification and propose algorithm being wrong. The results show that SVM-HOG algorithm is applied to the dataset then accuracy increase 8.75% as compared to PCA. Three performance metrics CMC, EPC and ROC are used for numerical value of accuracy.

(Karasawa 2017) and colleagues did a multi spectral object study for autonomous vehicles. There has been a fascination in the recent times by humans to develop a mobile robot that is autonomous vehicles that totally are autonomous with no human input. That requires the vehicle to be able to detect different type of objects like cars, people and various other means of transportation or objects. It is important for the mobile robot to be able to do that in both day and nighttime. Authors here looked at multi spectral images.

Multi spectral images include RGB images, near infrared images, middle infrared images and far infrared images. There are many objects that are difficult to recognize with RGB image, they can be detected through far infrared images by this method. This type of study is unique, and it requires the dataset that contains a lot of multispectral images. That can be trained. This type of data doesn't exist. The authors then decided to make their own dataset. A new pipeline called multispectral ensemble detection pipeline that will fully utilize the features of the images. There are two parts of the pipeline single spectral detection and ensemble part.

In the first experiment results show that every component is useful for the multi spectral object detection. In the second experiment which is the ensemble part of the pipeline entire system of multi spectral detection is evaluated. Mean average precision (MAP) is used as a metric for evaluation the results show that multi spectral object detection is 13% higher as compared to object detection based on RGB alone.

(Zhang 2013) did a study in which an improved method of Histogram Oriented gradient (HOG) was developed. This method took the baseline HOG method and then enhanced it to represent the information edge of the images. Future more a background subtraction method was used by the authors to track changes in real-time. The dataset used for this study was INRIA pedestrian dataset in which training data was 2416 and testing data 1126 positive sample. The resolution of images stood at 96 x 160. The detection window in the experiment was set at 96 x 160 with orientation number being 9 for static images.

For static pictures the resolution was 400 x 256. The average detection time was 94ms while time is 120ms. This approach improves the efficiency of detection by 22 percent. The second experiment was done on object tracking dataset which was video having a resolution of 640 x 480 and 30 frames every second. For processing a total of 96153.7 ms were used and per frame it took 178ms. The results of this were positive like the static image. But they weren't found out to be good enough for the algorithm to be deployed in real time.

Overall, the method improves the computational time required for human detection and object tracking. In this approach we see that in each dataset of images. Every region in an image or frame in a video is not equally important. This is why it is important to consider the non-uniform grid of points perspective. As shown in this paper to focus on only the informative parts of the image and ignore the less informative ones.

(Felzenszwalb 2010) and colleagues did a detailed study on object detection with discriminately trained parts-based models. This approach takes into consideration mixtures of multiscale deformable parts model. This method was tested on datasets in the PASCAL challenges. Before this the value of deformable parts-based models hadn't been quite established yet. Because they were never used on a benchmark dataset like PASCAL which is often used in competitions. In this method instead of labelling the dataset completely and supervising it. The researchers partially labelled the dataset. For data mining purposes a latent SVM model is used. Which combines a marginal sensitive approach with hard negative examples.

MI SVM is enhanced to turn it into a latent SVM. Once the training examples are specified (positive examples) and trained, latent function becomes convex. Originally latent SVM is semi convex. This all results in a training algorithm which is iterative in nature. It uses fixation of latent values and optimization of latent SVM objective function alternatively. The results of these models show highly accurate results on difficult to pass benchmark datasets. The proposed models can represent object classes that are highly variable. This framework can be extended into more latent structures. Parts by parts or hierarchical deep parts and mixture models that have many components are good examples of this.

(Liu 2021) and colleagues did a study for improving the speed and accuracy of a target variable in the detection of objects. Their approach was to present a framework which will improve target detection based on Yolov4 algorithm. The network framework's objective here is to lower the parameters of the model without compromising too much on accuracy. This will then result in the increase in speed of detecting the object. In the first step up sampling and down sampling of PANet links are strengthened by increasing attention mechanism CBAM. This will then improve the algorithm's ability to recognize occluded objects.

Additionally, a new convolutional called 'depth separable convolutional' is introduced to increase the speed of algorithm at the same time reducing the model parameters. Se-Net attention mechanism is added in the CSPDarknet53 a residual model, which will help in more attention by algorithm. Lastly Soft-NMS is added for optimizing the screening of frame being detected during detection process. The datasets used for the algorithm comparison are VOC2007+VOC2012dataset. Both these datasets are used for comparing state of the art models. Because these are very difficult to pass. Total number of images used are 27089 with 20 different types of objects included.

The experiment results show that Yolov4 does a better job than the proposed model with 93.1% accuracy based on CSPDarkNet53. But the speed of algorithm is higher in proposed model and accuracy is 1.9% lower for the Im-CSP-PANet. This is due to the reduction in parameters. But when the proposed model was compared with Faster-RCNN on just VOC2007 dataset model accuracy was 84.26% higher than any other modified and enhanced version of Faster-RCNN or CenterNet. With accuracy being 74.08% and 80.25% respectively.

2.3 Traffic Congestion

The main problem that researchers in the field of ITMS wan to tackle is traffic congestion. Two hurdles mentioned before traffic flow prediction and image processing and classification. Both indirectly look to tackle the problem of traffic congestion. Although there are other challenges associated that need to be tackled as well. A study done by (Bother and Schiller 2021) tackles the problem of traffic congestion. They proposed an algorithm called Multiple Routes evolutionary algorithm (MREA). This algorithm doesn't help directly in bottleneck area where traffic is stuck. Instead, this offers alternative routes to the drivers who ae willing to take. If a route towards a certain road is congested and traffic authorities offer the same alternative route to all the drivers stuck on that road. Then the next route will also be blocked, and nothing will change. That's why for this algorithm to perfectly work some of the drivers will have to take sub optimal routes as well. Sub optimal routes meaning that a few of the cars will be suggested a route different to the optimal route (which minimizes the time) but still the other route suggested will be closer to optimal. This kind of strategy was named by the authors as ‘strategic routing’. The results showed that the MREA can find the solution to problem 99.69% of the time while the time taken is only 40% of optimal solution.

All the models and methods that deal with the problem of traffic congestion first take a threshold level and then determine whether traffic is congested or not? This creates problems for two reasons. First one being that if there are a lot of vehicles flowing in one direction. Then that could be detected as being congested. But that’s not necessarily the case. Second if the road or a lane is singular, and all the traffic is headed to one direction then due the lane being one road being smaller could be detected as being congested. This means that almost all the cases congestion is treated as a binary option. Either there is congestion or none. To tackle this problem (Hossain and Adnan 2020) developed a method. The authors present a mathematical model that quantifies average speed over time on a road to check for the intensity of traffic. Figure 3 shows how the density slowly keeps increasing on road A while on road B it slowly increases but then decrease again rapidly. The evaluation metric was a congestion value. The value was calculated by considering residual values and the average road speed at different timeslots. This approach was found to be effective in detecting congestion earlier and it also helped in propagating traffic before it got congested. The dataset was taken from an urban center in which taxi trajectories were recorded.

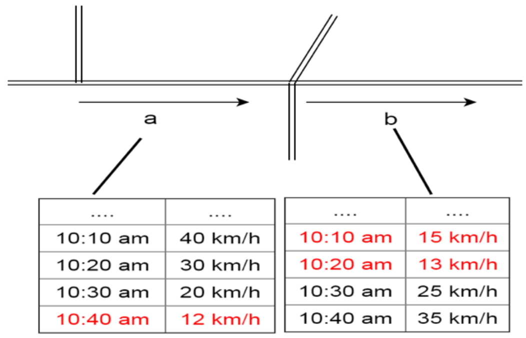


Figure 5. Avg speed in 30 mins time window

A study done by (Xiong and Vahedian 2018) predicted traffic congestion based on historical footprints in data. This is a different approach because it is predicting congestion in advance. And trying to tackle it before it even happens. The authors develop an algorithm called PPI_fast. The approach is called propagation graph approach. Like other algorithms studied so far, this one also takes a graph approach. This approach was tested on real world dataset from Shenzhen, China. The results showed that the PPI_fast algorithm predicts the future traffic propagation with higher accuracy as compared to the baseline brute-force algorithm. And the run time of proposed algorithm is also faster.

(Long 2008) and colleagues developed traffic congestion propagation model. Which is based on cell transmission model (CTM). The purpose was to identify bottleneck congested areas in urban road traffic networks. The reason why bottlenecks are more important to identify than the predicting the traffic flow is because that is what restricts the traffic efficiency and becomes a hindrance and act as sticking points. The model purposed by these authors is one that is linked. And it flows through nodes which connects all the ends with each other and helps in propagation for links. For estimating bottleneck area where traffic is congested a new estimation model is developed called average journey velocity (AJV). The model was tested on Sioux falls urban road traffic network. The authors used purposed model by simulating different traffic scenarios which then helped in identification of congested bottlenecks.

The simulation results show that increase in traffic continuously makes is complicated to predict the bottleneck areas where traffic is congested. To determine whether a link in the simulation model will become part of the congested area or bottleneck area or not depends on the current position of the link. Which in other words means it depends totally on the human factor. What does the person drive feel right at the time. It is just like predicting human behavior accurately. It is complicated and difficult to know through simulation what route or lane will the person in the driving seat will take.

(Liang 2017) and colleagues developed a model for predicting traffic cascading patterns. The idea presented by authors is that traffic on any road depends and correlated with the traffic patterns on nearby roads. Making it important to look and predict the patterns of traffic cascading from one road to the other one. In the study instead of looking at the traffic patterns between different roads simultaneously. They observed the traffic on each road separately. From which they studied the traffic cascading patterns on those roads. In the paper they look at three main influences in traffic congestion form one road to the other.

The first factor is the implicit factor which means that when and where did traffic reach from one point to the other instead of focusing how it got there. E.g., if the road no. 1 is congested then road no. 2 will get congested afterwards. This shows that the patterns in both roads which were studied separately are dependent on each other. The second factor is the multiple source factor. In which there are two different parts one is Neighboring traffic and the second is surrounding environment. The first one shows how traffic from one road is connected to the previous one. And cannot be looked at independently. The surrounding environment factor shows how on a given road can be different shops that are more crowded than the previous ones or some other external reason which doesn't include traffic directly. This will make it difficult to study the cascading patterns.

The third factor was geo-spatial correlations e.g., road 1 and road 5 are linked to each other but they aren't linked directly. They are part of the same network. Road 1 can impact road 3 traffic directly by congestion. But whether it will get to the fifth part of the network meaning road 5 is a challenge. And is difficult to integrate into calculating cascading patterns. This is called geo-spatial correlation. The authors used real world data sources like taxi trajectories, Poi's, meteorological data, and road networks. The results showed that the model developed by authors outperform the bassline models like NetInf and MultiTree. The score for the purposed model was 0.415 which showed superiority over sophisticated models like STC-DBN.

(Liu 2011) Did a study for discovering spatio-temporal causal interactions in traffic data stream. The authors proposed a model for detecting the outliers in spatio-temporal traffic data. The saw outliers as a reason not being able to predict the traffic more accurately. The algorithm they proposed developed causality trees based on the outlier data's spatial and temporal properties. They first build the graph of the region for which the outliers were to be detected. Region partitioning, formulating transitions, and Generating links were three main parts in constructing these graphs.

In region partitioning they looked at the major roads of the city. The model deployed for image segmenting was Connected Components Labelling. This was going to divide the map. The reason for this is that subdivision is of a polygonal region is known to be NP complete. By formulating transitions, they scanned the trajectory of taxis in one of their datasets where they had trajectories of taxis over a period. Every trajectory was then put in sequence of whichever pair of regions it belonged to. By generating links, the algorithm observed that what links are connected to each other. Which means that the links observed the origin region of the taxis and the destination. Based on this it makes further calculations. The result after applying the algorithm shows that anomalies in the Beijing traffic data.

(Nguyen 2017) and colleagues developed an algorithm that discovers the patterns in traffic congestion. Just like mentioned in the previous study before that detection of traffic patterns can become a complicated challenge to tackle because there are multiple influences on each road. The goal of authors in this paper was to develop an algorithm that will construct causality trees from the areas where traffic is congested. Then based on the spatial and temporal information it will calculate their propagation probabilities. The reason for such an extensive approach is because the authors thought that the existing literature hasn't investigated the propagation patterns and the casual relation among traffic from different roads properly.
In experiment causal congestion pairs were developed meaning pair of two roads. After which the snapshot of five minutes was captured. That was necessary to see whether there is a direct relationship of traffic on one road to the other. The proposed model STC gave higher accuracy as compared to STO when compared to 20 pairs for causal relation. The data on for testing the algorithm was real-time collected. Which included over 8000 observations. Sampling interval was averaged for 5 minutes for each pair. Number of sites were 281 in total and 586 segments in the entire network.

(Mandayam, 2014) developed two model for tackling the problem of traffic congestion. One was a deterministic fluid model which was based on their previous work of conservation laws. Second is a mean field model. It is based in infinite server of queues making at a series model. The idea here is to make stages in the server look like highways. The model accounts for different times the vehicles in each dataset arrived on the highway and exited that highway. The model takes the map of a given highway in ‘time-varying function’. Here both the models do the job equally well in obtaining the highway map.

There is cost accounting parameter in the functions that captures the total amount of extra time spent on a highway due to congestion. Which doesn’t include the time spent driving. All this goes into creating an optimization problem which is convex. It helps in determining how to transfer traffic from peak hours to off peak hours to help the traffic flow better and helping in reducing congestion. The results of these models show that if 10% of the highway traffic before peak hour is transferred to off peak hours. Which in this instance is 15 minutes before peak time starts then cost of congestion is reduced by 19%.

(Wang 2023) and colleagues did a research study to develop a congestion cause discovery system. Which aims to discover the reason for traffic congestions and delays. The previous ways discussed depend a lot on human factor and what the driver decides at a given time. This is why it becomes difficult to predict the congestion and keep track of its pattern accurately over the period. There are few challenges like low availability of data that knows the reason for congestion, other factors like complex spatio-temporal relations and other relations that effect the traffic but are unknown.

This is why a two-module system was developed by authors first one being congestion feature extraction module. This system extracts the important features from the data distinguishing different causes of congestion from the given data. Second is a deep semi supervised learning model which discovers the causes of congestion with limited labeled data. For pre training the model it learns the limited information before entering training stage. The next stage is clustering data. Where there are additional two stages. First one aims to produce soft clusters by using classic deep embedded cluster model. The second one tries to group the features extracted from the pre trained model by using k means cluster. Then use the results to fine tune the network.

The results of the experiments show that the proposed model is superior to state of the are methods. It demonstrates significant improvement over other models. Unsupervised models like K means, DEC and DCN. And semi supervised models like DAC, DTC and MCL were also compared. The data used for testing was made up of road segments. In which highways war divided into segments of 100 to 200 meters.

(Jian 2012) and colleagues proposed an image processing model that detects the traffic congestion levels by reviewing the CCTV cameras feed. The main purpose for developing this approach was to make an algorithm which will help in traffic congestion classification in the developing world. Because the infrastructure for roads and CCTV tends to be of lower quality in the developing world as compared to the developed world. This algorithm will help with the images that are of lower quality, low resolution and noisy. Due to this the researchers observed after looking at the CCTV live feed form Brazil and Kenya congestion collapse behavior extends congestion for extended periods.

To take on this problem of long-lasting congestions the algorithm developed looked towards a ‘local decongestion protocol’. Which after being integrated with traffic feeds can prevent local congestion and stop the burst of congestions which results in longer traffic delays. The sources of data are Rio de Janeiro, Brazil, Mombasa, Kenya and Nairobi, Kenya. After feeding the data to the simulator, running experiments through proposed algorithm and simple network topologies. The results show that local de congestion model enhances the road network capacity in a localized road setting. This is a very effective model because it is low cost and next step in this can be to calculate the road traffic density in real time and take the algorithm one step further form there.

2.4 Intelligent Incident Detection

The incident detection is an important challenge that needs to be overcome in the future smart cities and modern urban centers. This does not mean that the goal is to take the incidents happening down to zero. That is not possible at all because there is no way possible to avoid collusion between vehicles or even objects on the roads altogether. Even after making all the vehicles fully autonomous. As shown in figure one there are two type of incidents (1) Planned (2) Unplanned. The planned one can said to be maintenance of roadside, sports events, or other events like peak hour congestion (not exactly planned but frequent travelers on the road know this in advance). Meanwhile the unplanned can be classified as spilled oil or other material from trucks, accidents, or another similar emergency. Now we will look at the literature for intelligent incident detection systems. In this project we will look at the AI techniques that are deployed and have been deployed in the recent years for incident detection. Some of those models are Fuzzy logic (FL), Support Vector Machine (SVM), Regression Methods (RM), Neural Networks (NN), Naïve Bayes (NB), Hidden Markov Model (HMM) etc.

(Ahmed and Hawas 2012) developed a model based on threshold level that will look at real time data and then classify the incidents in urban road networks. The purpose of this project was to develop a model that will look at a blockage even on a single lane. The threshold values were based on traffic volume combination, timing of signal and a ‘pre-timed signalized intersection area’. The authors used NETSIM simulator. The results were positive and the purpose for the study was completed.

Another project done by the (Ahmed and Hawas 2013) used fuzzy logic in real time incident detection on urban road networks. The data for this was collected by the detectors fixed on signal. The same simulator same as previous one was used. The results showed that fuzzy logic model can be used in incident detection.

(Gosh and Smith 2014) proposed an algorithm that was for signalized urban arterials. The authors chose four SVM, Multi Layered Feed Forward Neural Networks (MLFNN), Fuzzy Wavelet Radial Basis Function Neural Network (FWRBFNN) and Probabilistic Neural Networks (PNN). The data was simulated on VISSIM simulator and then it was used as input on which the AI models were applied to. The results showed that MLFNN model works best among the four tested model. And this should be implemented in the signalized areas.

(Yang and Lin 2009) developed a technique that that included SVM classifier and FL algorithm to detect incidents. The data for this approach was fusion data. Which means fixed detectors and prove vehicles. They concluded that if fusion data is used as a source the outcome is better to detect incidents.

The unplanned many times are caused by rash driving or driving that is not legal i.e., driving above speed limit or turn on roads that aren't legal. (Akoz and Karsigil 2011) used continuous HMM to driving that deviate from normal behaviors.

(Chlyah and Dardor 2016) presented a model that detects urban roads that are signalized. The data was taken form inductive loop detectors (IDL's). The data is then processed and classifies by multi agent system and support vector machines (MAS and SVM). SUMO simulator was used for the validation processed. The results were good and in acceptable range.

(Ahmed and Hawas 2016) presented a binary logit model for incident detection in urban road network. This was a new approach compared to their other two studies before. Again, they used

NTESIM simulator and selected a random incident. Some threshold values were decided before training. After the inputs were fed results showed good accuracy and conclusion was that these models can be deployed to detect incidents.

(Boumhidi and Hatri 2017) used a fuzzy deep learning model for incident detection in intersections. The data was fed to the SUMO simulator. The data gathered was from ILD's. The authors applied a fuzzy deep neural network learning. Stacked auto encoder was used for processing the fed data. For determining how efficient this method is the results were compared to other ANN.

(Gu and Qian 2016) used a different approach for detection rather than directly relying on ILD's or other source to gather data. They turned to twitter feeds to look to identify real time incident detections. The authors used a semi naïve bayes classification model. The tweets were classified in two categories (1) traffic accidents (2) non traffic accidents. The results were then extracted by processing and filtering tweets.

(Pan and WU 2017) presented a different approach for detecting incidents. They relied on mobile sensors to detect traffic accidents. They used Vehicular ad hoc networks (VANET) to classify what constitutes as abnormal traffic behavior like speed violations, lane changing without proper signaling etc. Then after feeding data SVM method was used, and the results were positive indicating that this technique can be used for traffic incident detection.

(Dardor and Bouhmidi 2018) did a study in which they combined to different approaches. First, they wrote a genetic algorithm and then combined it with the SVM. This method was developed to detect urban traffic incidents on 'original urban network scenario' and 'Freeway like scenario'. In the genetic algorithm there were overall nine steps. Which first generated initial set of population. Then checked the best fit. Update results and select chromosomes (chromosome is randomly chosen candidate in population). Then perform crossovers and mutate. At the end it returns chromosomes that have the best fit. Then the dataset was divided into train and test. The train dataset was 70% while the test dataset was 30% of total data. SVM neurons are then trained. For simulation purposes SUMO simulator was used. The results show that when genetic algorithm combined with SVM. We get better results in classification accuracy and false alarm rate is lower. The performance was evaluated by comparing proposed model with others.

CHAPTER Three

Models and Algorithms

In this chapter we will extensively analyze various machine learning and deep learning models that are commonly used in time series analysis and image classification for forecasting and object detection. Not only just the mathematical and statistical computations that are behind developing these models and algorithms but also, we will apply these models on real world traffic datasets and images. For traffic congestion and incident detection models and algorithms with the highest accuracy will be selected. After which best models from all the components will be put together to be integrated in intelligent traffic systems.

3.1 Traffic Flow Forecasting

Traffic flow forecasting can be done by using the machine learning and deep learning algorithms that help in the prediction of sequential data. Traffic data is sequential data as at any given time of the day traffic is increasing or decreasing in a sequence. For this purpose, five key and most used timeseries models are selected:

i. Auto Regressive Integrated Moving Average (ARIMA)
ii. Vector Autoregression (VAR)
iii. Prophet
iv. Extreme Gradient Boosting (XGBoost)
v. Long Short-Term Memory (LSTM)

Now, we will construct each one of them individually and then apply them on real world datasets to compare the accuracy of each one of these models.

3.1.1 Auto Regressive Integrated Moving Average (ARIMA):

The Autoregressive Integrated Moving Average (ARIMA) model is a one of the most widely used time series forecasting method used for making models and forecasting time series data. ARIMA is made up of three key parts: Autoregressive (AR), Integrated (I), and Moving Average (MA). (Box and Jenkins 1976) first presented this model in their works Time Series analysis: Forecasting and control. Let's break down the mathematics of each component:

Autoregressive (AR) Component:

The autoregressive part of any ARIMA model presents the relationship between the current value of the time series and its past values. It is written as ARp, p here is said to be the autoregressive order of the series. The equation for ARp is as follows:

$$Y(t) = \alpha + \beta_1 * Y(t-1) + \beta_2 * Y(t-2) + \ldots\ldots + \beta i * Y(t-i) + \varepsilon(t)$$

Yt represents current value of the time series at time t.

$\beta_1, \beta_2, ..., \beta_p$ are the autoregressive coefficients for lags 1 to i.

εt represents the error which is also called white noise at any given time t.

α is the intercept. It can also be called as a constant term in this equation.

Integration:

The integration in any ARIMA model is done so that the time series can be made stationary. Because most of series are not stationary differencing is necessary. Taking the difference makes the variance constant. It is denoted as d, where d is the order of differencing. The differencing operation is performed to remove trends and make the time series stationary.

$$Y't = Yt - Y(t-d)$$

Y't represents time series after differencing.

Yt represents time series before differencing.

d represents the order i.e., how many time the series has been differenced.

Moving Average (MA):

The moving average part in any ARIMA model represents the relationship between the current value in the series and error terms form the previous series which can be called white noise. It is written as MAq, where q represents moving average's order. The equation after including MAq is as follows:

$$\mathbf{Yt = c + \delta t + \mu_1 * \delta(t\text{-}1) + \mu_2 * \delta(t\text{-}2) + ... + \mu q * \delta(t\text{-}q)}$$

Yt in the above equation represents value of the time series at time t.

δt, $\delta(t-1)$, ..., $\delta(t-q)$ represent the error terms which can be called white noise terms at time t and lags 1 to q.

μ_1, μ_2, ..., μq represents Moving Average coefficients of the lags 1 to q.

c represents the constant term which can be also called intercept.

The ARIMA model in summary can be written as ARIMA (p, d, q), where these alphabets are:

p represents autoregression order.

d represents differencing order.

q represets moving average order.

Parameters in the model (β and μ which are also called coefficients) are calculated by using a commonly used statistical method called Maximum Likelihood Estimation (MLE). The model afterwards is used for predicting the values in future time for any given series by using these parameters.

3.1.2 Vector Autoregression (VAR):

In 1980 Christopher Sims came up with a new model from sequential data analysis forecasting called vector autoregressions. The purpose of developing this initially was to build a macro-econometric model which will increase the predictive accuracy of existing autoregressive models. Which in turn will decrease the error in macroeconomic variable forecasting (Stock and Watson 2001). Breakdown of complete model is as follows:

Vector Autoregression (VAR) is a commonly used model in statistics for analyzing the relationships between multivariate time series. It is a model built by extending the single variable or univariate Autoregressive Model (AR), where the relationship between a single time series variable and its past values are modeled. VAR deals with multivariate time series variables at the same time giving higher predictive accuracy.

For instance, consider a VAR(p) model, where p represents the total number of lags in the model. In this VAR(p) model where represents k time series variables, each variable can be called a linear combination of its own previous values and the previous values of all other variables in the series. Computationally VAR calculates the current value as weighted average of current k series based on its previous values **($B_1, B_2, ..., B_k$)** can be denoted as follows:

$$\mathbf{B_t = c + X_1B_{(t-1)} + X_2B_{(t-2)} + ... + X_p*B_{(t-p)} + \varepsilon_t}$$

Where:

B_t represents a column vector of the k time series variables at time t.

c represents column vector of constant terms for each of the k equations.

$X_1, X_2, ..., X_p$ represents matrices of coefficient of dimensions k x k for lags 1 to p.

ε_t is a vector of column for error that can be called white noise terms, in which the assumption is that they are normally distributed where mean is zero and covariance is constant.

The coefficient matrices X_i capture the current and lagged relationships between the multiple variables. These are the matrices that are to be estimated from the series.

For estimating parameters (c and the matrices of coefficient X_1, X_2, ..., X_p), usually different methods like Bayesian methods, maximum likelihood estimation (MLE) and least squares estimation (LSE) are used. After the parameters have been estimated, one can use the VAR model for various purposes, including forecasting future values of the time series variables, impulse response analysis, and Granger causality testing, among others.

The p order in the VAR model, what variables are to be included, and various other decisions in the modelling process should be made after thoroughly exploring data.

3.1.3 Prophet:

Fb Prophet is a time series forecasting model designed by Facebooks core data science team in 2017. It was first presented by (Taylor and Latham 2017). Breakdown of complete model is as follows:

It's designed in such a way so that it can handle seasonal patterns in data one or more than one, specifically for time series. While the model itself is quite sophisticated and computations involved in it are beyond the scope of this project. We must look at the overview of components involved in the model. The Prophet model uses a Bayesian approach, and the complete details of the model are quite extensive.

The description of components and example equation is as follows:

Additive Decomposition: Time series in Prophet is decomposed into three main components: trend, seasonality, and holidays/effects.

Trend Component: The trend is usually in this model is a piecewise linear or logistic growth curve. It's represented as a function of time t and is written as g t. Changepoints are used to identify where the trend changes direction.

Seasonality Component: Weekly and yearly seasonality's both are accounted for in Prophet. Fourier series represents these seasonal patterns. The yearly seasonality can be represented as a function of time as s1 t and the weekly seasonality as s2 t.

Holiday/Event Effects: This part is more subjective if the user has information about some specific holiday the information can be passed into the model.

Error Component: The error term, ε t, is the noise or residuals in the data. The assumption here is that it is normally distributed, and the mean is zero.

Forecasting Equation: The forecasting equation in Prophet can be written as:

$$Yt = g(t) + s1(t) + s2(t) + \ldots\ldots + sn(t) + \varepsilon(t)$$

Values observed at any given time t are represented by Yt, trend component is represented by g(t), s1(t) is the yearly and s2(t) is the weekly seasonality component. And sn(t) represents any holiday seasonality component.

Model Fitting: Bayesian approach is used in Prophet for model fitting that incorporates uncertainty in the seasonal and trend components. Markov Chain Monte Carlo (MCMC) is used for sampling or optimization techniques to estimate the model parameters.

Prediction: After fitting the model it can be used to forecast the future values. Prophet generates point forecasts as well as uncertainty intervals (prediction intervals) to capture the uncertainty in the forecasts.

3.1.4 Extreme Gradient Boosting (XG Boost):

XG boost in recent times has become one of the most popular machine learning models in time series forecasting. It is being widely used in financial markets now. It was first presented by (Chen and Guestrin 2016) in University of Washington.

(Leventis 2018) provides a detailed description of XG boost model:

XGBoost objective function: XG boost's aim is to minimize the objective function. It's a combination of different functions. The objective function here can be said as a function within a function. To minimize the loss, it has a loss function and then a regularization term. The loss function is used to calculate the error between the dependent or target variable (y) and predicted values (ŷ) of those variables. It can be represented by (y, ŷ). For regression purposes mean square error (MSE) is used while the classification tasks use log-loss (cross entropy).

To tackle the problem of overfitting of data. The L1 or known as Lasso and L2 or know as Ridge regularization are used. λ represents parameter for controlling the regularization in equation.

$$\mathbf{Y^t = \sum_{i=1}^{n} \boldsymbol{l}(y, \hat{y}^{(t-1)} + f_t(x_i)) + \lambda(f_t)}$$

Y^t is the objective function's value.

l is the CART learner.

(y, ŷ) are representing target variables values and its predicted values.

$f_t(x_i)$ represents the changes in the value of input from one time t to the next time in the series.

λ is the regularization parameter.

Gradient Boosting Trees: XG boost is specifically designed for time series and sequential data analysis. That's why its ensembles decision trees sequentially. Every tree in the sequence corrects the error of the previous tree by fitting a new tree to the loss of the previous tree's prediction. New trees can be represented as follows:

$$\mathbf{-\sigma * \varphi\ (loss) / \varphi\ (previous\ value)}$$

σ is the learning rate. This is the most important part of the equation as it accounts for the step size. After which new tree will be built.

φ is the parameter that represents the error and previous value.

Approximate Greedy Algorithm: XG boost uses an approximate greedy algorithm. It is built upon the basic exact greedy algorithm. The job of this algorithm is to find the best split in tree learning. The problem is that this is extremely computationally heavy. So, approximate greedy algorithm builds on this (Chen and Guestrin 2016). There are two ways to perform this algorithm one is the global variant and the other is local variant. The global variant splits the trees initially in tree construction process. Meanwhile the local variant proposes it again and again after every step. The comparison of both done by authors on Higgs bason dataset showed that local variant requires fewer candidates (fewer trees).

$$\sum_{i=1}^{n} \frac{1}{2} \mathbf{hi(ft(xi) - gi/hi)^2 + \lambda(ft) + constant,}$$

h_i represents each data point.

gi/hi squared represents the weighted square loss.

λ is the regularization parameter.

Pruning: XG Boost has tree pruning methods which mainly removes any tree that doesn’t helps in reducing the error. Which in turn helps in lowering the number of trees that we need for the prediction.

Weighted Quantile Sketch: Due to heavy computational background of this model. It requires weighted quantile sketch to be used which calculates the statistics of features when the trees are being built for predictions.

3.1.5 Long Short-Term Memory (LSTM):

This model was first developed by (Hochreiter and Schmidhuber 1997). The Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN). It consists of various gates. Which capture the long-term dependencies in data. It tackles the problem of vanishing gradients that occurs in traditional RNN. LSTM is widely used in various tasks, including natural language processing, time series analysis, and more. Any problem in which the solution depends on sequential data analysis LSTM's can be used for that.

LSTM consists of several different parts. These are usually called gates. It has an input gate, output gate and forget gate. It also contains a cell state which is responsible for the memory of LSTMs. These all states work with each to control the flow of information through the cell state. Let's break down the mathematics of LSTM's:

Input Gate: Input gate is responsible for the passing on the input from the current time x(t) to the next cell state. Which can be also called Hidden state. Because both cell state and hidden state are dependent. Input gate is the part where which information flows to next part is decided. It also extracts information from past hidden state (h{t-1}). The value for any given output from this gate is between 0 and 1.

$$\mathbf{i(t) = \sigma(W(i) * [h\{t\text{-}1\}, x(t)] + b(i)}$$

σ is the activation function.

i(t) represents the output form the input gate.

W(i) represents the weight form matrix.

b(i) represents the bias matrix.

Forget Gate: The forget gate is responsible for discarding the information from the previous cells. It also gives a value between 0 and 1.

$$\mathbf{f(t) = \sigma(W(i) *[h\{t-1\}, x(t)] + b(i)}$$

f(t) represents the output form forget gate.

W(i) represents the weight form matrix.

b(i) represents the bias matrix.

σ is the activation function.

Cell State: There are two ways to look at cell state once before and the next after it is updated. In the first part it updates the cell by taking information from current time t and the value from past time. After which new cell is produced which follows the same process all over.

Mathematically it can be represented as follows:

$$\mathbf{C(t) = tanh(W(i) *[h\ (t-1, x(t)+b(i))}$$

C(t) represents the output for the cell state.

Tanh is the activation function.

W(i) represents the weight matrix.

b(i) represents the bias matrix.

The second part is when the cell is updated by combing the information form past value of cell states. Output from forget gate because this passes on only the key information and discards the irrelevant part. And the candidate for the cell state C(t). Mathematically this can be written as follow:

$$\mathbf{C_t = f(t) * C_{t-1} + i_t + C(t)}$$

C_t here represents the cell state after it’s been updated.

Output Gate: The output is responsible for providing the information from current cell state to the next one. It produces a value between 0 and 1. This is done by taking current time’s input x(t) and past value of hidden state (h{t-1}). Below is the formulation:

$$O_t = \sigma(W(i) *[h [t-1, x(t)] +b(i))$$

O_t represents the output form this gate.

σ is the activation function.

W(i) is the weight matrix.

b(i) is the bias matrix.

Hidden State: The ending part is where the updated cell state and the output for output gate are put together. Which gives the result form one unit of LSTM.

$$H_t = O_t * \tanh (C_t)$$

C_t is the cell state after it is updated.

H_t is the hidden state.

Tanh is the activation function.

This all description is from a single LSTM cell. Typically, multiple LSTM units are put together with each other. The calculations of which can be extremely extensive and done by computers only.

3.1.6 Methodology and Data:

i) Methodology

The five models mentioned in the previous sections ARIMA, VAR, Prophet, XG Boost and LSTM. All will be applied one by one. And then compared to each other on five different datasets. The process is to first collect the data and then clean it. Only variables relevant to the traffic are to be included. The data for the forecasting must be real time recorded data because real world data is most relevant to this project. After cleaning data will be split into test and training. Then each model will be fed the data for processing. The results will in the end be presented in a table for all five models to be compared. Below is the complete visual representation of the data.

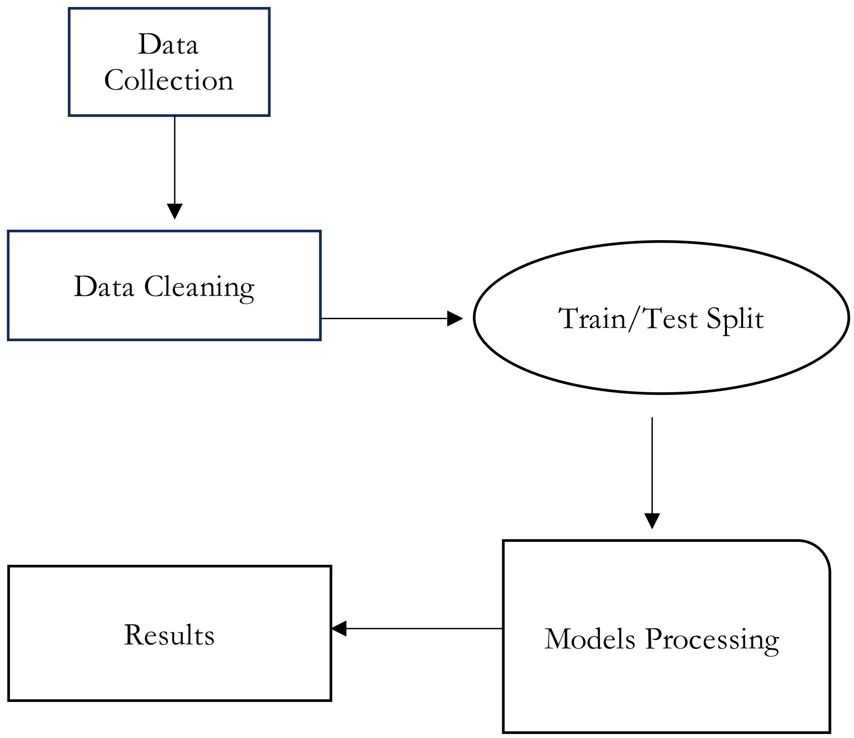


Figure 3. Methodology to be followed.

ii) Data

Data is taken from five different sources overall. All the data the is collected is taken from credible sources. The links for the sources of data will be in the appendix section. All these sources deployed techniques for data collected in real time.

The first source of data is Taxi data from Beijing, China. It contains trajectory of more than 10000 taxis. The data is based on week. Feb 2, 2008, to Feb 8, 2008. The data is collected from GPS trajectories. The overall trajectories reach a total of 9 million kilometers. We will be looking at only a select few taxi ids. Running the entire data set isn’t the scope of this project. As it is used more for city planning or private use purposes.

The dataset number two is from Perth, Australia It is from 2014 to 2018 average weekly data. Some of the data is average on two weeks intervals. The data was collected form GIS services by the city. The dataset number three is from the San Jose city Average daily traffic data. The is total from 2018 to 2022. It is having average daily traffic, average daily one way and average daily two-way traffic.

The dataset number four is from California State Highway Network. It is also the same is previous one it collects average weekly data. Traffic is counted by electronic instruments. It is not the data from one specific highway but an average of all the highways. Which is done by moving instruments from one place to another. At the end total daily traffic is average. There are more than 7000 objects represented in the data. Not all are cars. It includes motor bikes, trucks, and other means of transportation.

The fifth and the last dataset is from Munich it is collected through induction loops. The dataset contains data from 6 crosses. In one of Munich’s urban areas. It accounts for a total of 56 days. Vehicles data is for every 5 minutes for 12 hours per day for 288 days for a total of 16128 objects passing through loops.

3.1.7 Experiment and Results:

The coding was done for experiments on the google colab environment. System used was MacBook Air 2017, processor 1.8 GHz Dual core Intel Core i5. Memory 8 Gb 1600 MHz DDR3. Graphics Intel HD 6000 1536 MB. TensorFlow and other libraries were downloaded by pip command all the code is attached for relevant sections in the appendix. Next is the results of the experiments. We will look at mean absolute error (MAE) of each model on all five datasets.

ARIMA ON Beijing Dataset results:

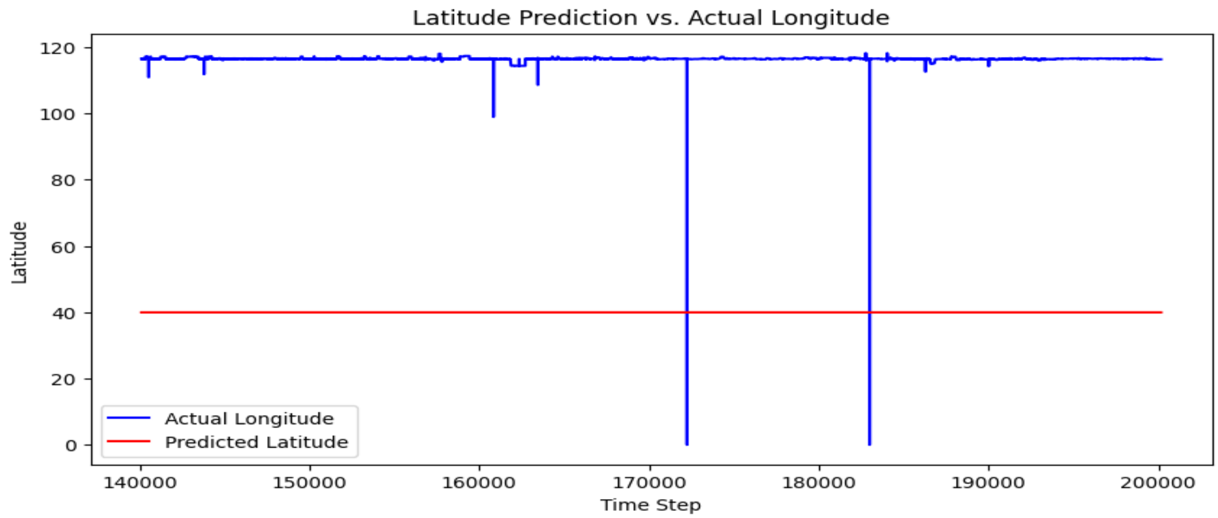


Figure 4 Longitude vs Latitude

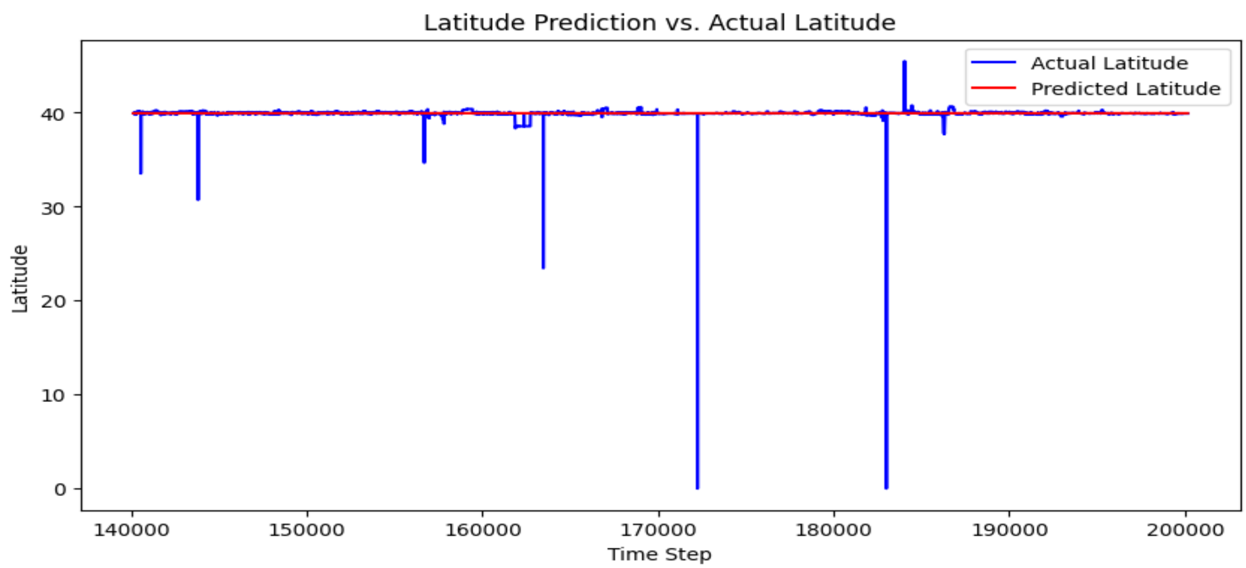


Figure 5 Time step vs Latitude

Figures 4 and 5 show the graphs for Beijing taxi dataset and trajectory the taxis follow over a period. The mean absolute error for ARIMA model is 0.10. After predicting latitude with longitude as independent variable. The train test split for every dataset followed is 70/30. T handle the missing values on every dataset. Knn imputation is used.

ARIMA model on California dataset:

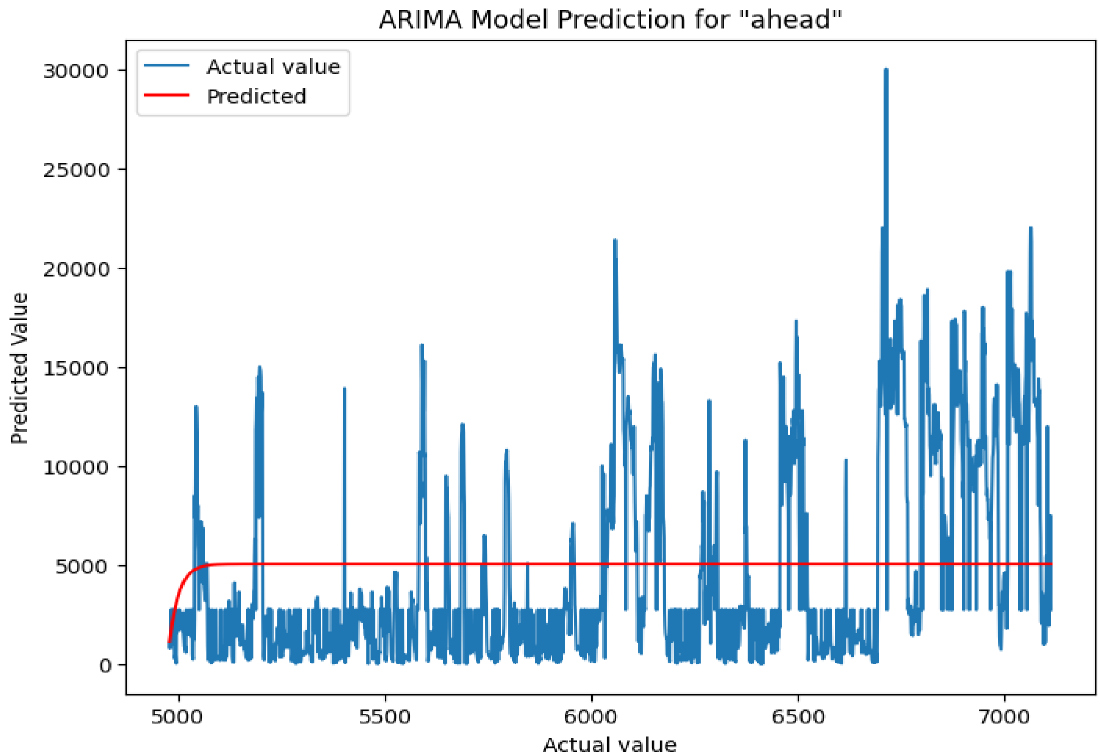


Figure 6. Average weekly traffic

Figure 6 shows the Actual vs Predicted value of average weekly data. Mean absolute error (MAE) of ARIMA model on this dataset is 4406.37.

ARIMA model on San Jose dataset:

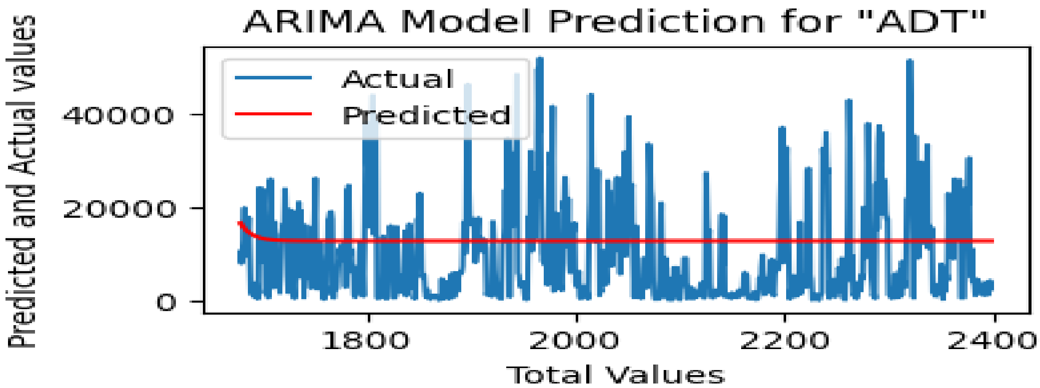


Figure 7. Average daily traffic

Figure 7 shows the average daily traffic of San Jose dataset Actual vs Predicted. Mean Absolute error (MAE) 9271.84.

ARIMA model on Perth dataset:

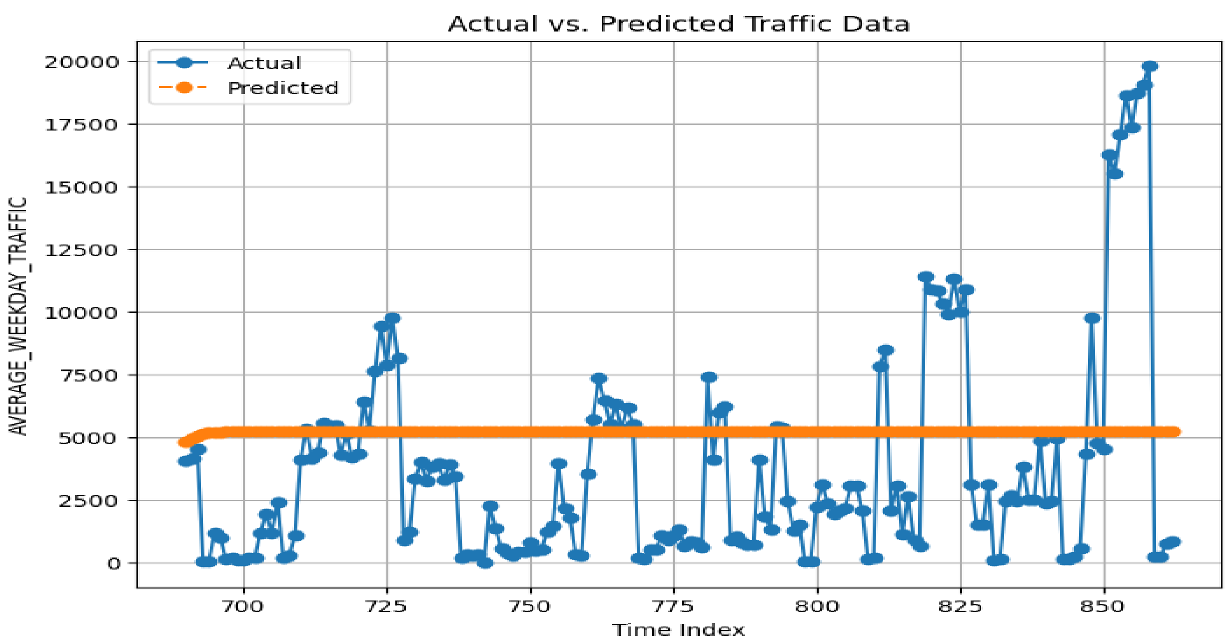


Figure 8. Time index vs Average Weekly Traffic

Figure 8 shows the Actual vs Predicted value of average weekly traffic on Perth dataset. Mean absolute error for the dataset is 3667.27.

ARIMA model on Munich dataset Firstcross:

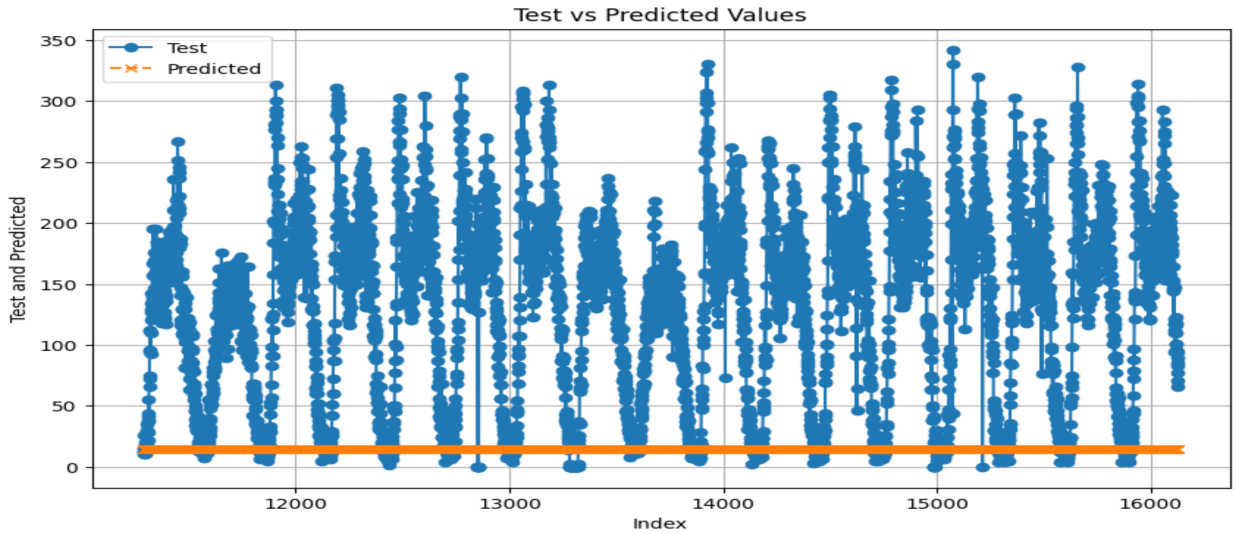


Figure 9. Time index vs Test and Predicted traffic.

Figure 9 shows the test meaning Actual and Predicted values for traffic on the Munich traffic datasets firstcross. Overall, there were 6 crosses but using dataset from two will serve our purpose. Mean absolute error (MAE) is 116.07

ARIMA model on Munich dataset Secondcross:

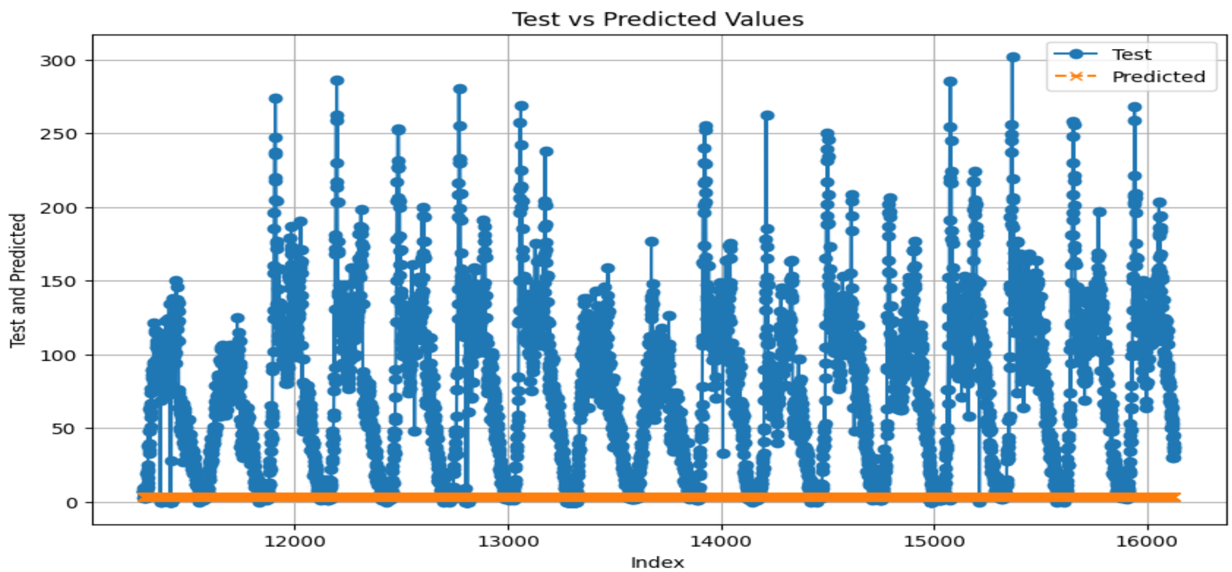


Figure 10. Time index vs Test and Predicted traffic.

Figure 10 shows the Actual vs Predicted values with time step being on the x axis for the second cross on our Munich traffic dataset. Mean absolute error (MAE) is 72.68. Next, we will look at the VAR model on all 5 datasets.

VAR model on Beijing Dataset:

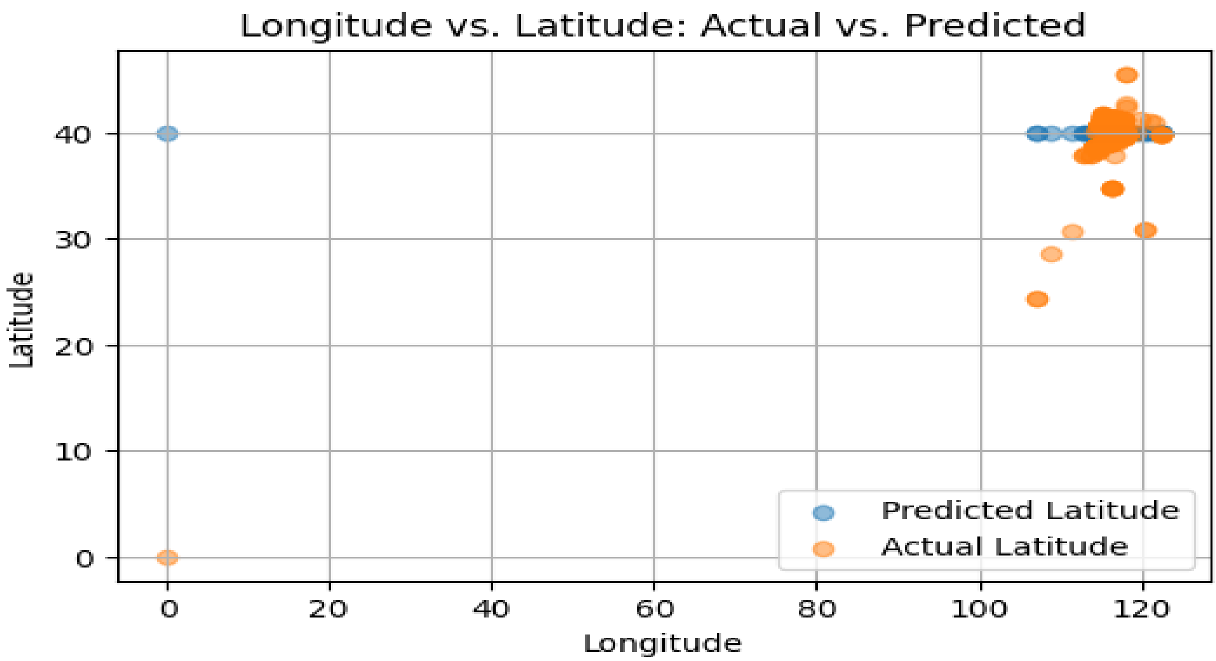


Figure 11. Actual vs Predicted Latitude.

Figure 11 shows the Actual and Predicted latitude. This model can be further enhanced, and more features can be added. Model can be then applied on a large enough dataset to see the traffic trajectory over time. Mean absolute error (MAE) is 0.094.

VAR model on California Dataset:

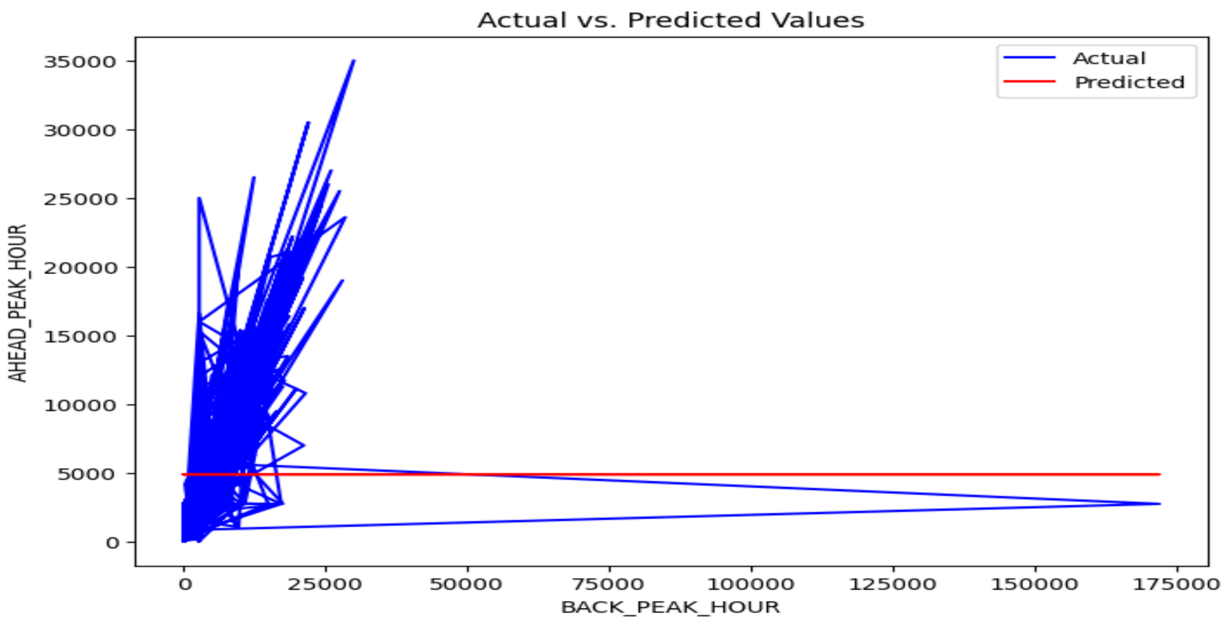


Figure 12. Actual vs Predicted values of Ahead_Peak_Hour

Figure 12 shows the graph of Actual and predicted values of California dataset using VAR. Mean absolute error (MAE) is 4110.06. This is lower than the previous model used.

VAR model on San Jose Dataset:

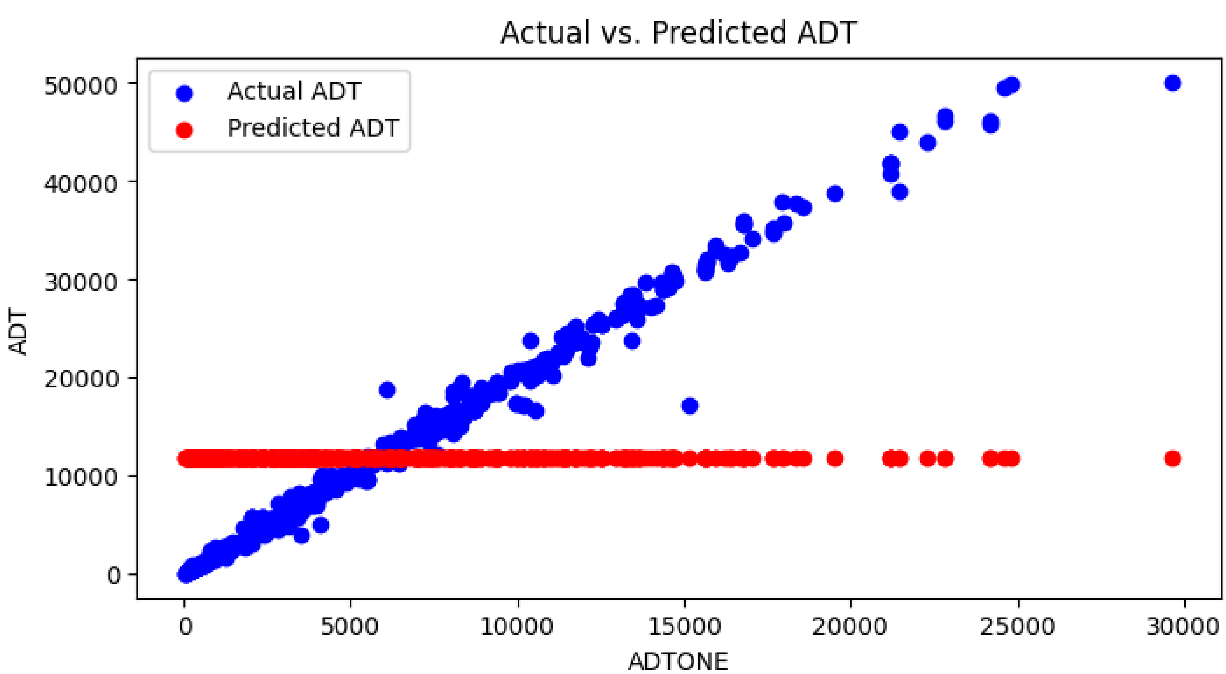


Figure 13. Actual vs predicted values of Average daily traffic.

Figure 13 shows Actual vs Predicted values of Average daily traffic on San Jose dataset using VAR. Mean absolute error (MAE) is 8167.40. MAE is improving for every dataset so far.

VAR model Peth dataset:

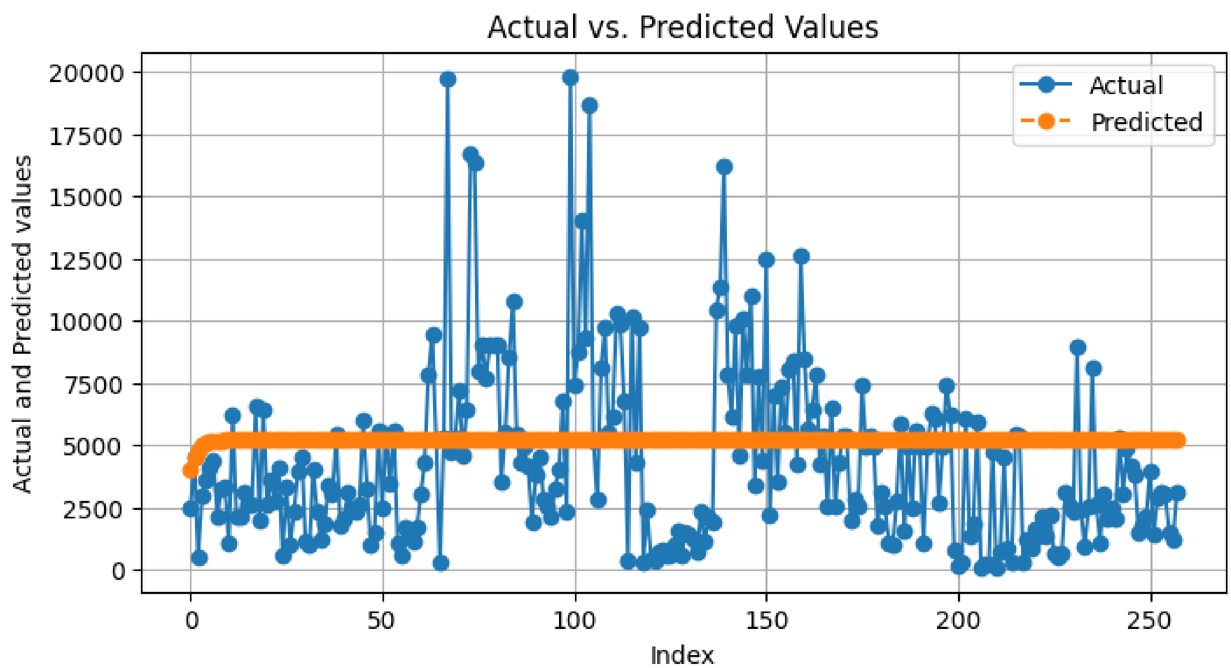


Figure 14. Actual vs Predicted values of average weekly traffic.

Figure 14 shows the Actual vs Predicted values of weekly traffic averaged on Perth dataset using VAR. Mean absolute error (MAE) is 2897.72. MAE is still improving using VAR on every dataset.

VAR model on Munich Dataset Firstcross:

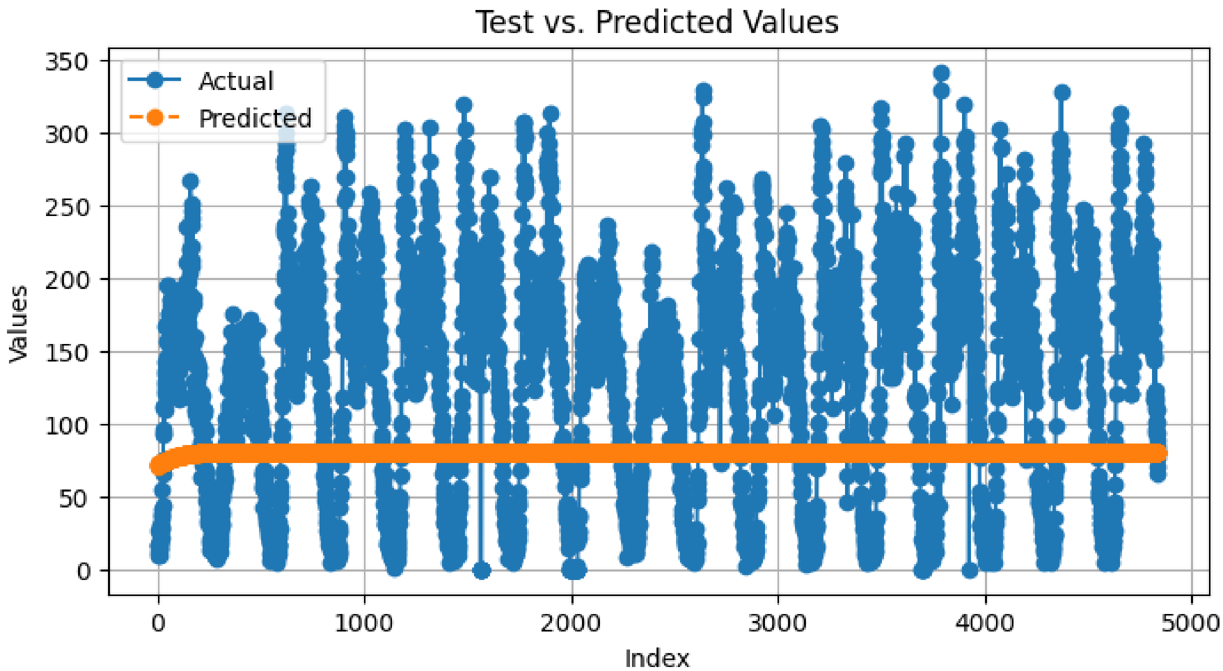


Figure 15. Actual vs Predicted values of Firstcross.

Figure 15 shows the Actual vs Predicted values of traffic volume on Munich dataset using VAR.

Mean absolute error (MAE) is 80.11.

VAR model on Munich Dataset Secondcross:

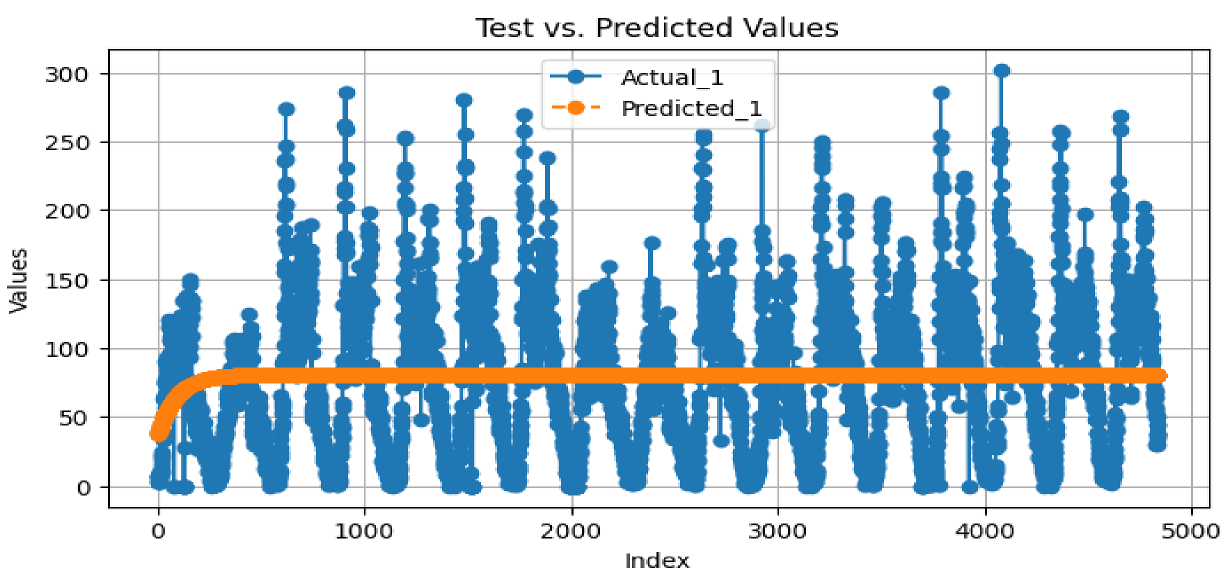


Figure 16. Actual vs Predicted values of Secondcross.

Figure 16 shows Actual vs Predicted values of traffic volume of Secondcross in Munich dataset. Mean absolute error is 46.48. The results improved compared to previous model on all the datasets using VAR compared to ARIMA. Making it a better model to be used for Traffic flow prediction.

Prophet model on Beijing Taxi Dataset:

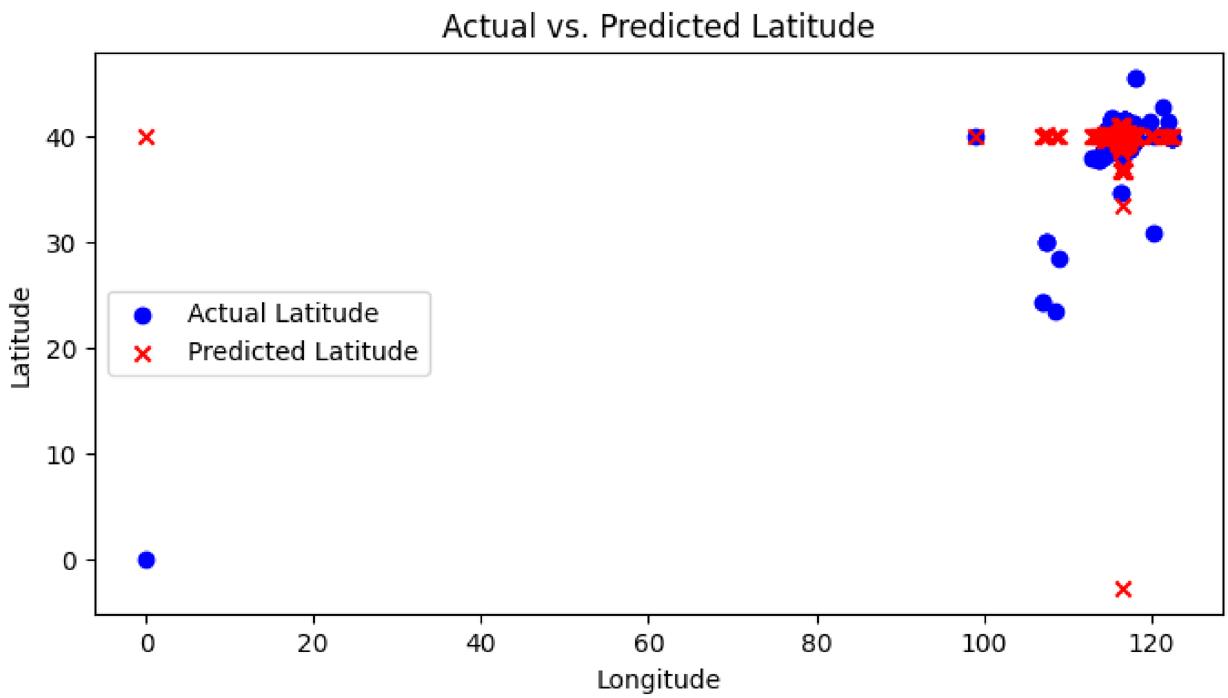


Figure 17. Actual vs Predicted values Latitude.

Figure 17 shows the Actual vs Predicted values of Latitude on Beijing Dataset when Prophet model is used. Mean absolute value (MAE) is 0.10.

Prophet model on California dataset:

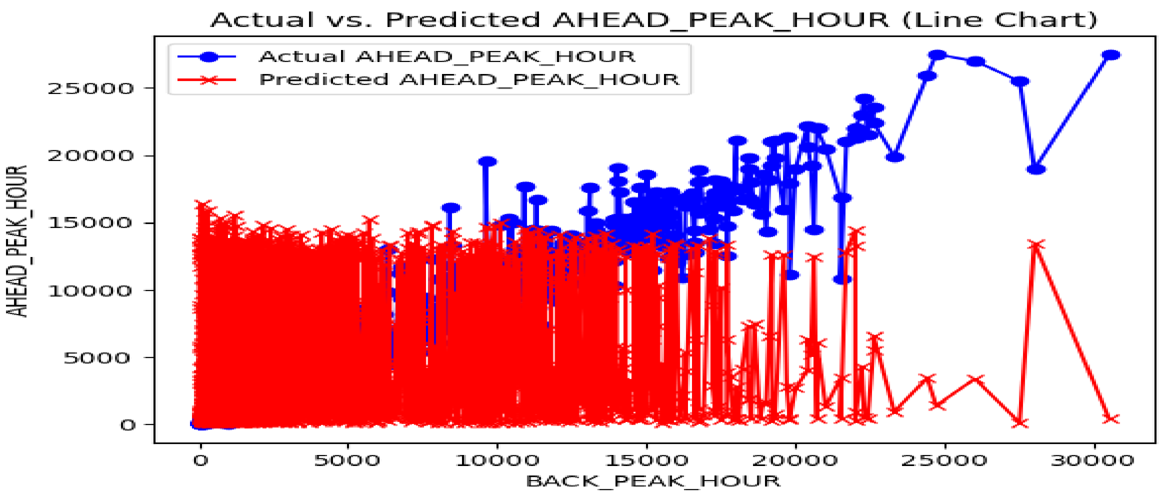


Figure 18. Actual vs Predicted values of Ahead_Peak_Hour

Figure 18 shows the Actual vs Predicted values of Ahead_Peak_Hour of California traffic using Prophet. Mean absolute error is 5241.83. This is higher than both the previous models ARIMA and VAR.

Prophet model on San Jose dataset:

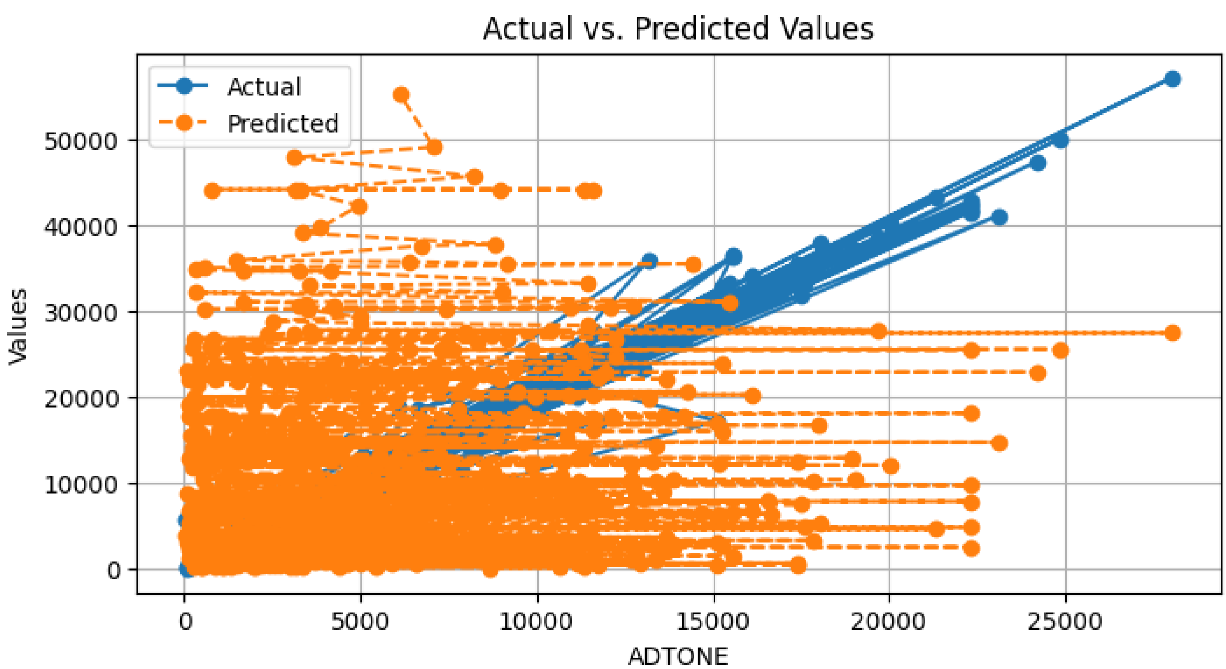


Figure 19. Actual vs Predicted values of Average daily traffic.

Figure 19 shows the average daily traffic Actual and Predicted values on San Jose traffic data using Prophet model. Mean absolute error is 10252.23. This is higher than both model ARIMA and VAR.

Prophet model on Perth Dataset:

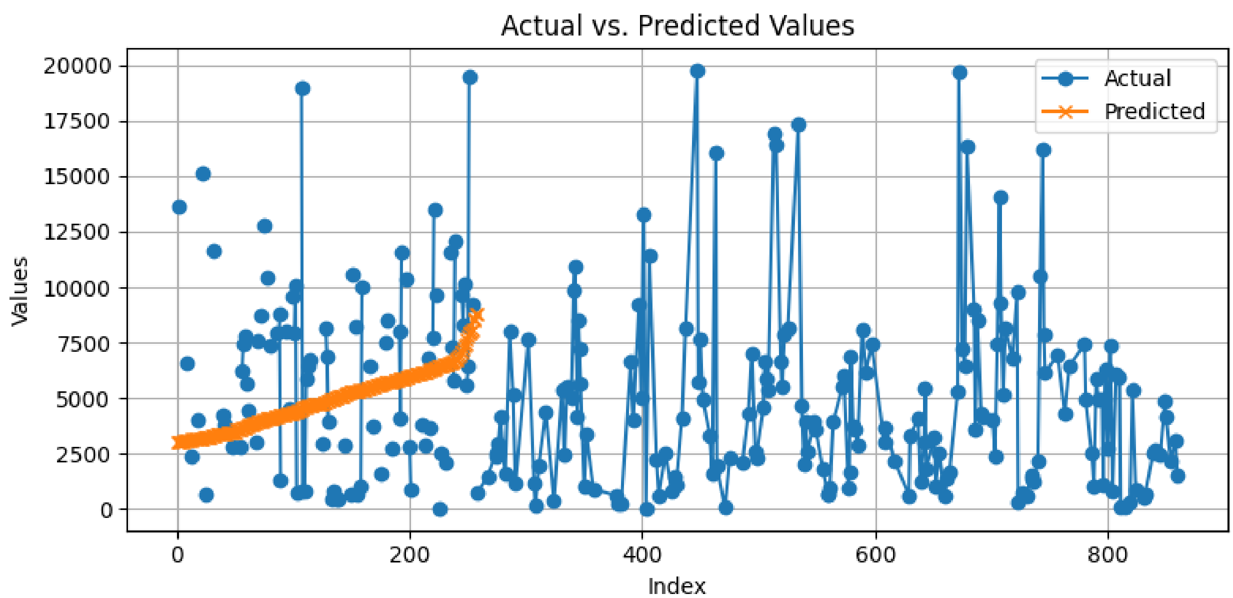


Figure 20. Average Weekly traffic Actual vs Predicted.

Figure 20 shows the Actual vs Predicted values of weekly traffic averaged on Perth dataset using Prophet. Mean absolute error (MAE) is 3290.29. This is higher than Higher than VAR but lower than ARIMA.

Prophet model on Munich Dataset Firstcross:

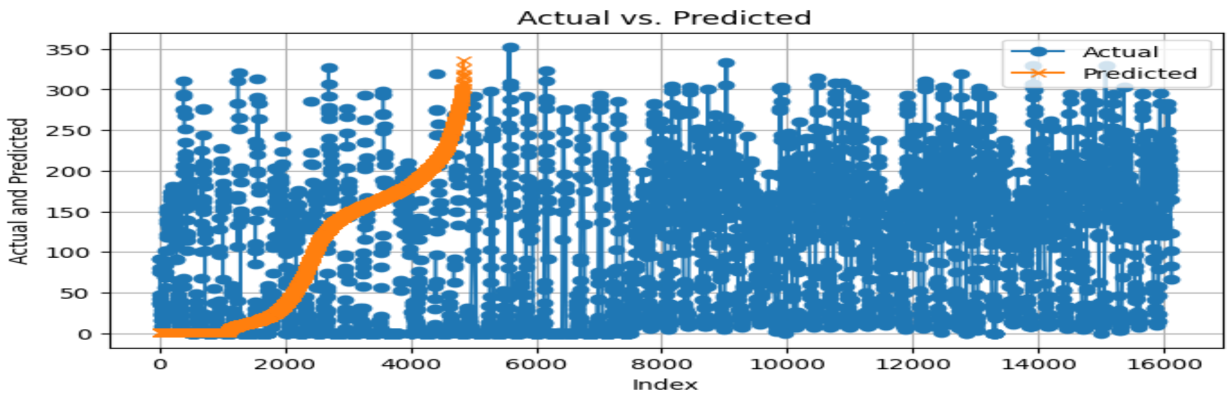


Figure 21. Actual vs Predicted values of Firstcross.

Figure 21 shows the Actual vs Predicted values of traffic volume on Munich dataset using Prophet. Mean absolute error (MAE) is 95.95. This is higher than VAR but lower than ARIMA.

Prophet model on Munich Dataset Secondcross:

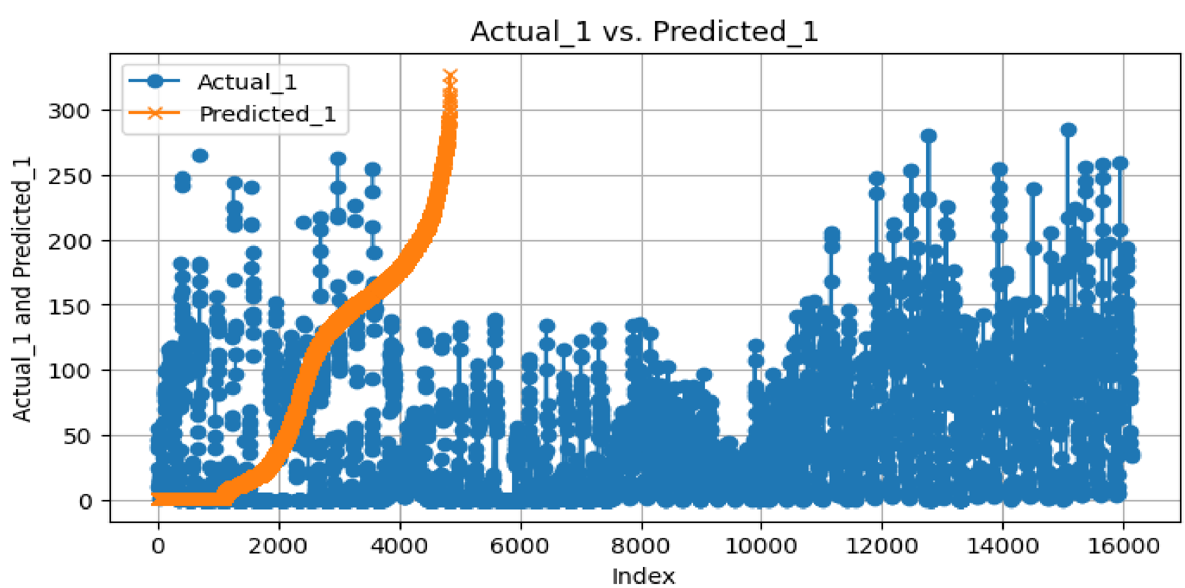


Figure 22. Actual vs Predicted values of Secondcross.

Figure 22 shows the Actual vs Predicted values of traffic volume on Munich dataset using Prophet. Mean absolute error (MAE) is 84.09. This is higher than VAR but lower than ARIMA.

XG boost model on Beijing Dataset:

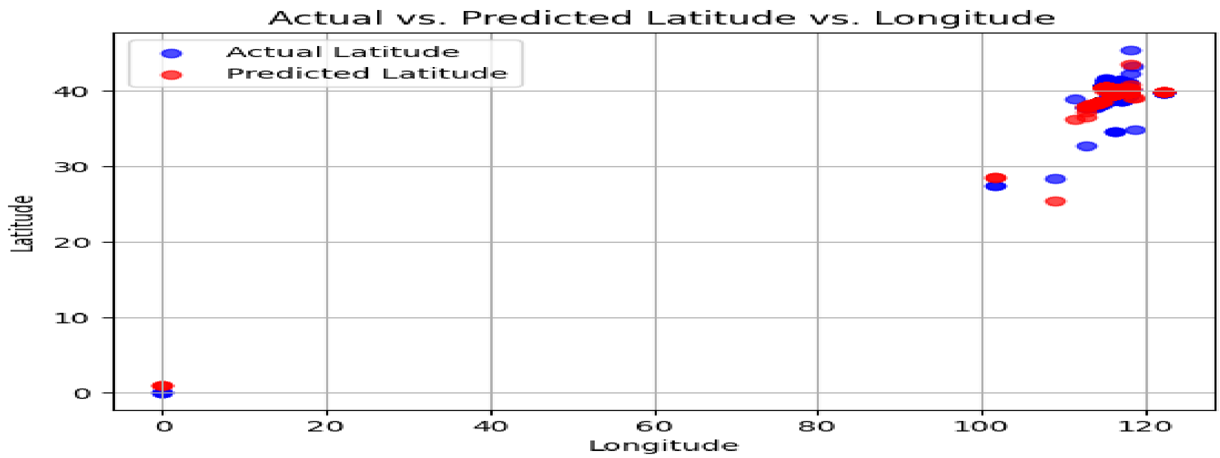


Figure 23. Actual vs Predicted Latitude.

Figure 23 shows the Actual and Predicted values of Latitude, Longitude being on x axis using XG Bost model. Mean absolute error (MAE) is 0.06. This is the lowest among all other models.

XG boost model on California Dataset:

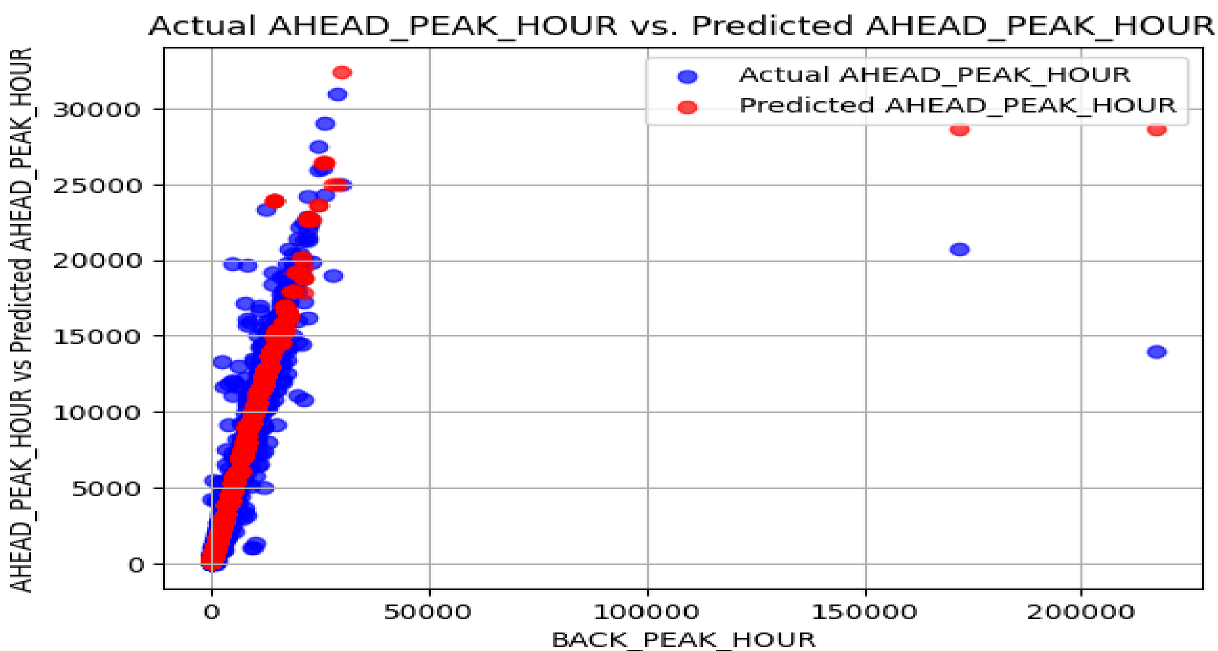


Figure 24. Actual vs Predicted values of Ahead_Peak_Hour.

Figure 24 shows the Actual vs Predicted values of Ahead_Peak_Hour using California dataset using XG Boost model. Mean absolute error (MAE) is 662.58. This is the lowest among all other models. If more features are extracted and new variables are added. This is likely to decline even further.

XG boost model on San Jose Dataset:

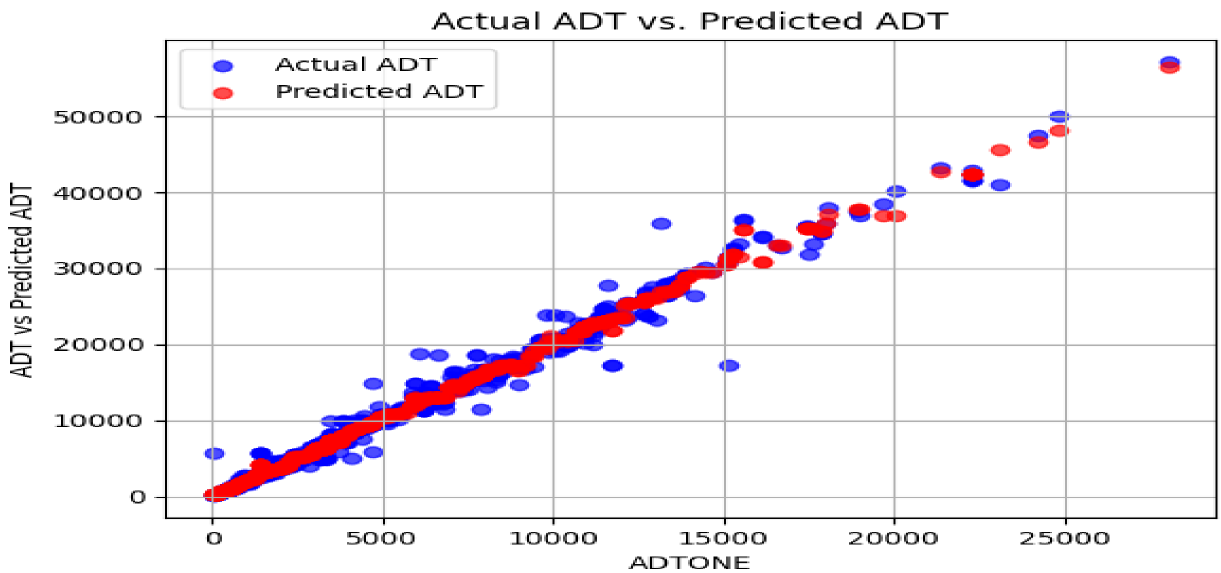


Figure 25. Actual vs Predicted values Average of daily traffic.

Figure 25 shows the Actual and Predicted values of average daily traffic on San Jose dataset using XG Boost. Mean absolute error (MAE) is 615.45. This is the lowest among all other models.

XG boost model on Perth Dataset:

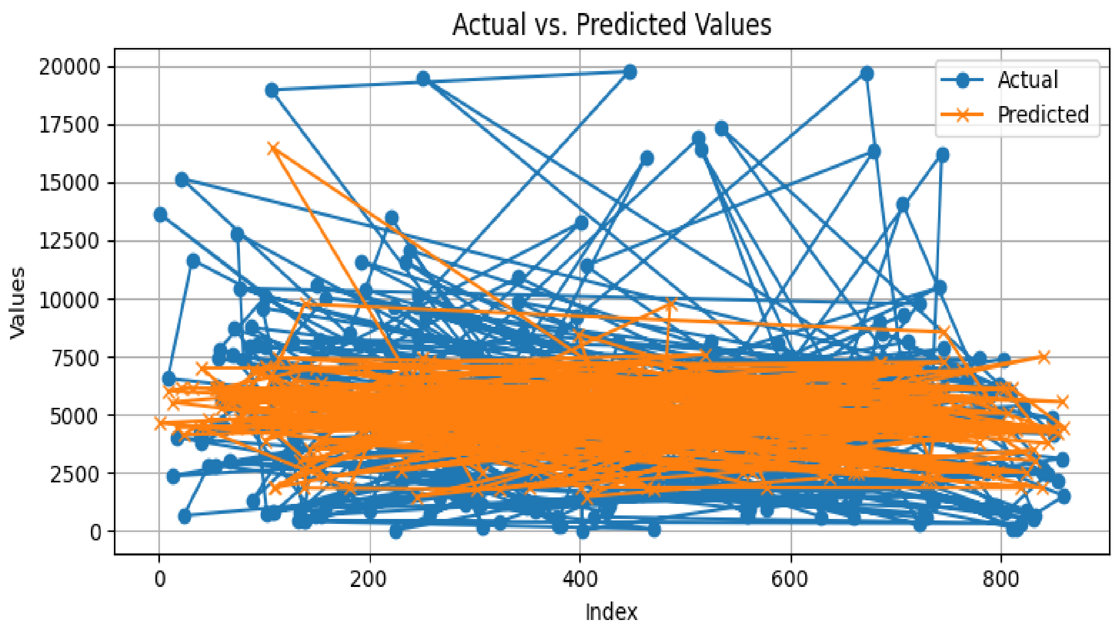


Figure 26. Actual vs Predicted values of Average weekly traffic.

Figure 26 shows Actual and Predicted values averaged in Perth form traffic dataset using XG Boost model. Mean absolute error is 3114.62. This is lower than ARIMA and Prophet but higher than VAR.

XG boost model on Munich Dataset Firstcross:

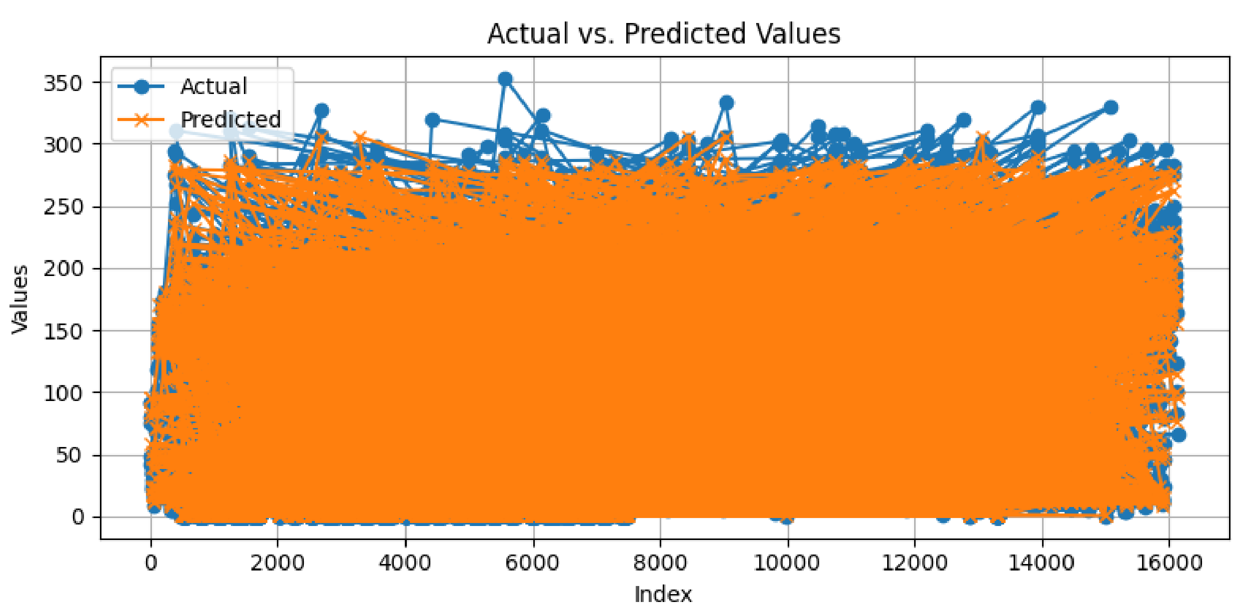


Figure 27. Actual and Predicted values Firstcross.

Figure 27 shows Actual and Predicted values of Firstcross in Munich dataset using XG Boost model. Mean absolute error (MAE) is 11.50. This is the lowest among all the models.

XG boost model on Munich Dataset Secondcross:

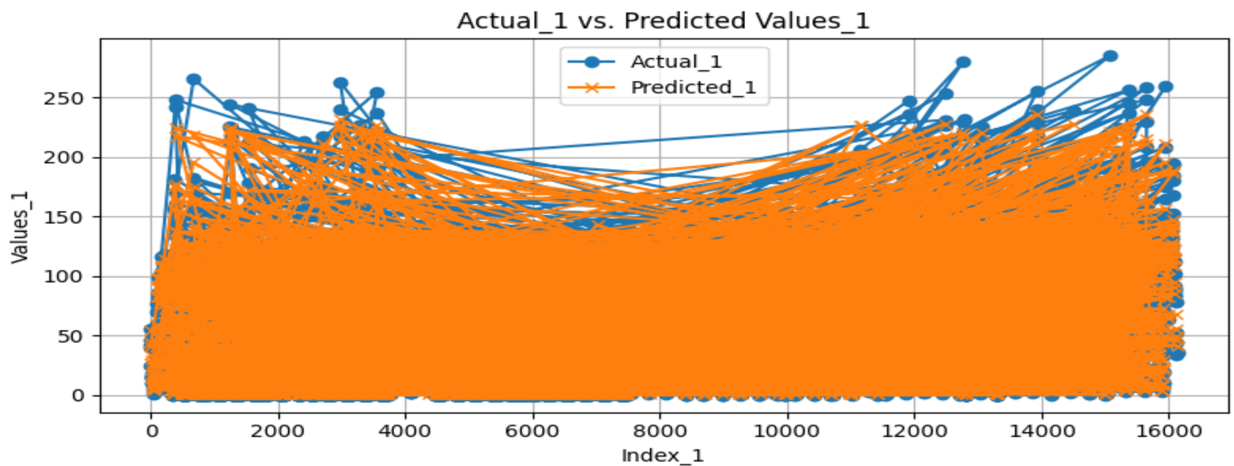


Figure 28. Actual vs predicted values of Secondcross.

Figure 28 shows Actual and Predicted values of Secondcross in Munich dataset. Mean absolute error (MAE) is 7.79. This is the lowest among all the models.

LSTM on Beijing dataset:

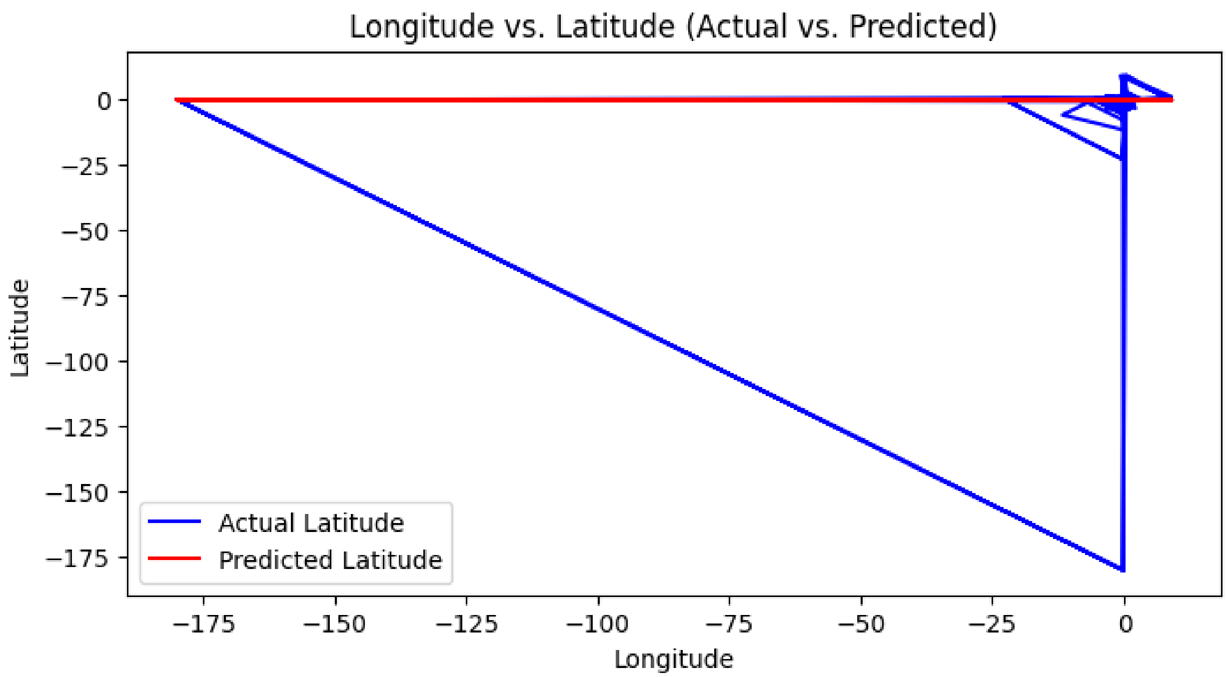


Figure 29. Actual vs Predicted values of Latitude.

Figure 23 shows the Actual and Predicted values of Latitude, Longitude being on x axis using LSTM model. Standard Scaler is used for scaling the data. Mean absolute error (MAE) is 0.23. These are the worst results among all the other models.

LSTM model on California dataset:

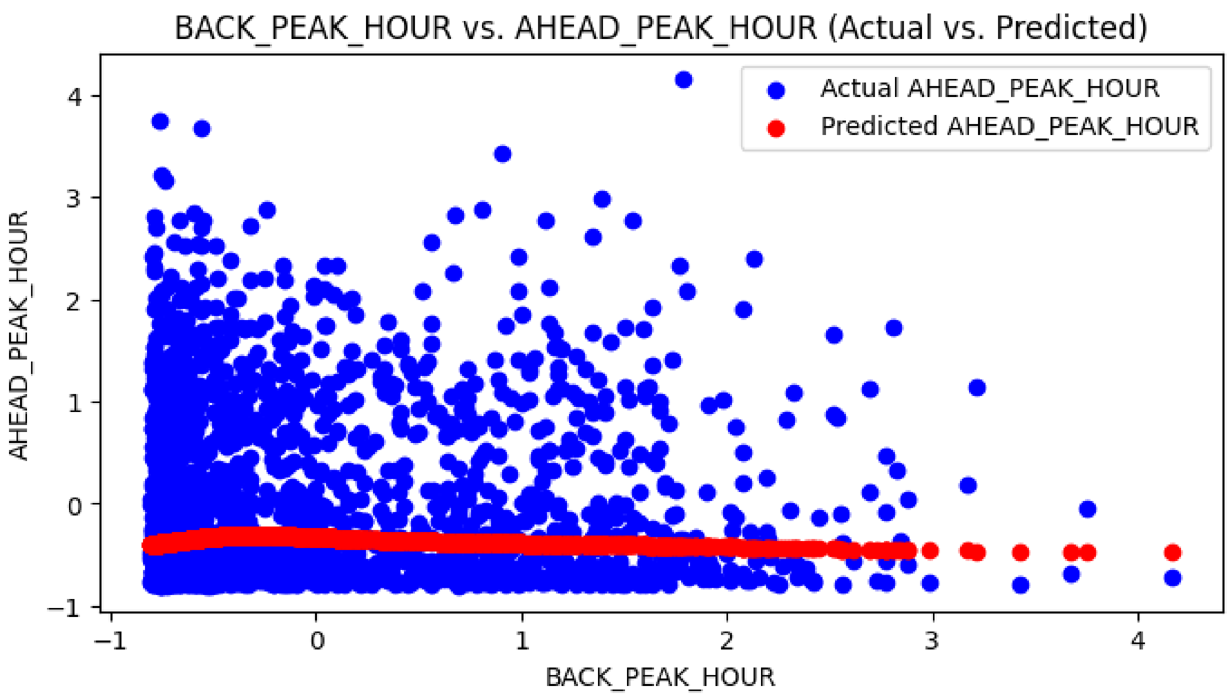


Figure 30. Actual vs Predicted values of Ahead_Peak_Hour

Figure 30 shows the Actual vs Predicted values of Ahead_Peak_Hour on California dataset using LSTM model. Mean absolute error (MAE) is 0.63. This is the lowest among all other models. This is the best among all the other model because of the scaling of data.

LSTM model on San Jose traffic:

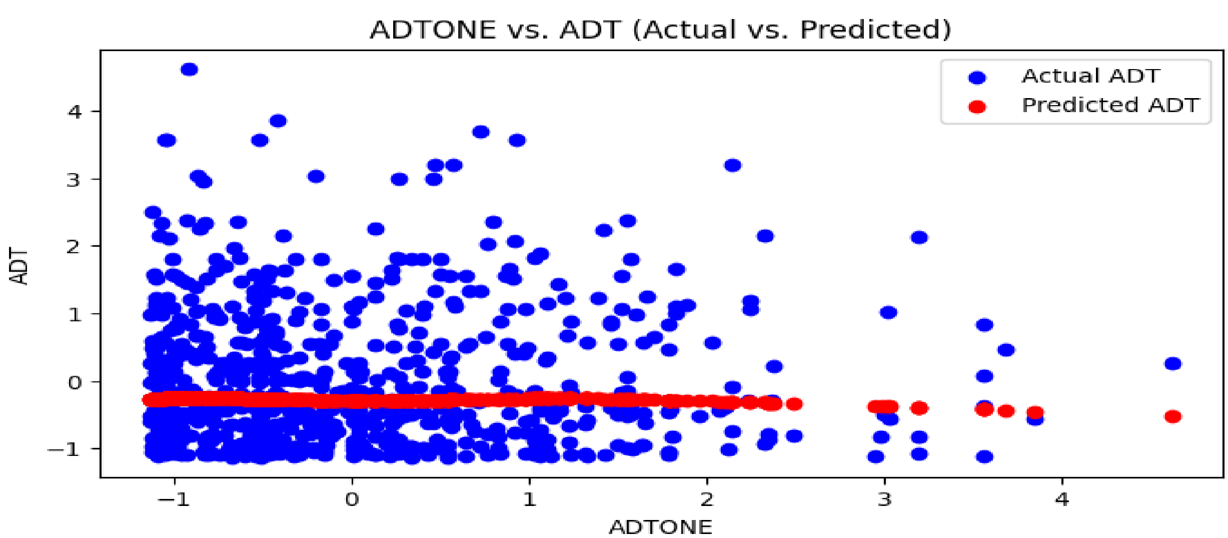


Figure 31. Actual vs Predicted values Average of daily traffic.

Figure 31 shows the Actual and Predicted values of average daily traffic on San Jose dataset using LSTM model. Mean absolute error (MAE) is 0.77. This is the lowest among all other models.

LSTM on Perth Dataset:

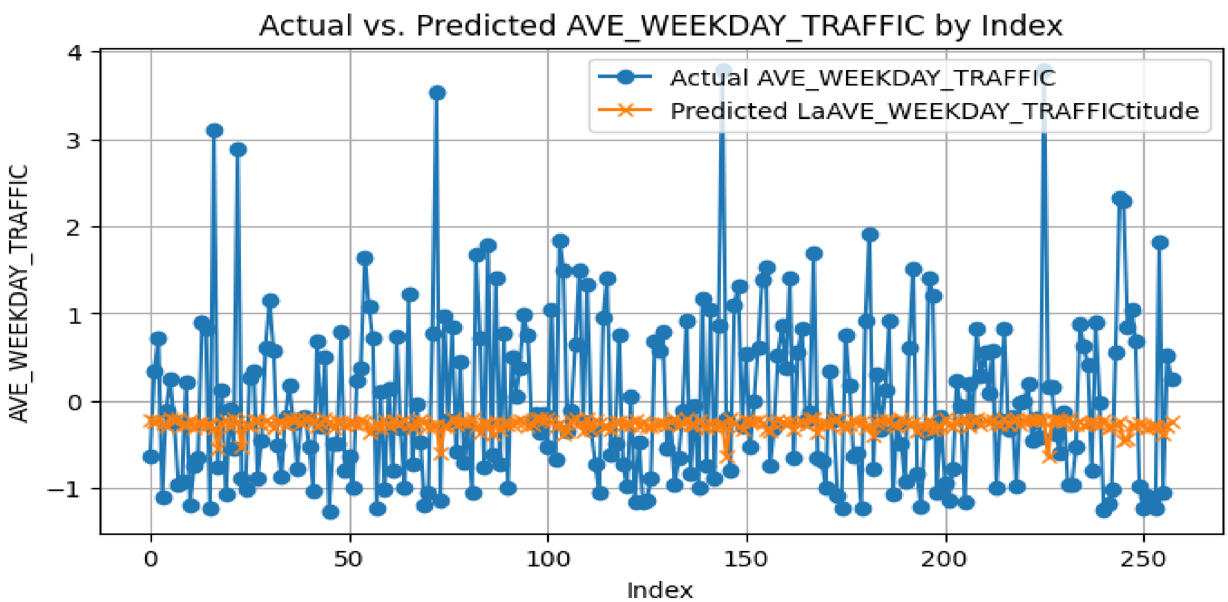


Figure 32. Actual vs Predicted values of Average weekly traffic.

Figure 26 shows Actual and Predicted values averaged in Perth form traffic dataset using LSTM model. Mean absolute error is 0.76. This is the lowest among all other models.

LSTM on Munich Firstcross:

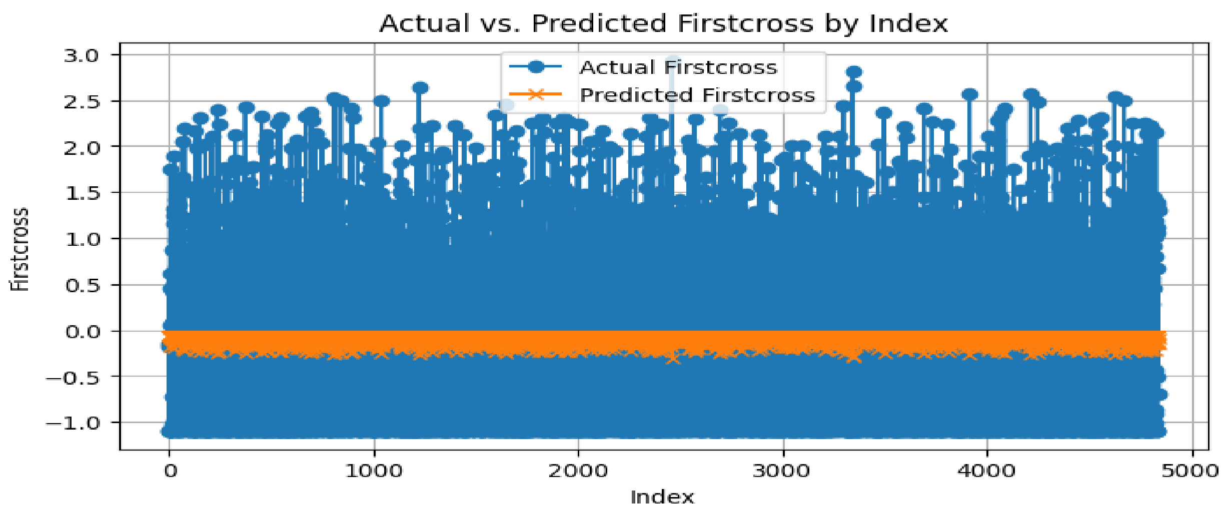


Figure 33. Actual and Predicted values Firstcross.

Figure 33 shows Actual and Predicted values of Firstcross in Munich dataset suing LSTM model. Mean absolute error (MAE) is 0.90. This is the lowest among all the models.

LSTM on Munich Secondcross:

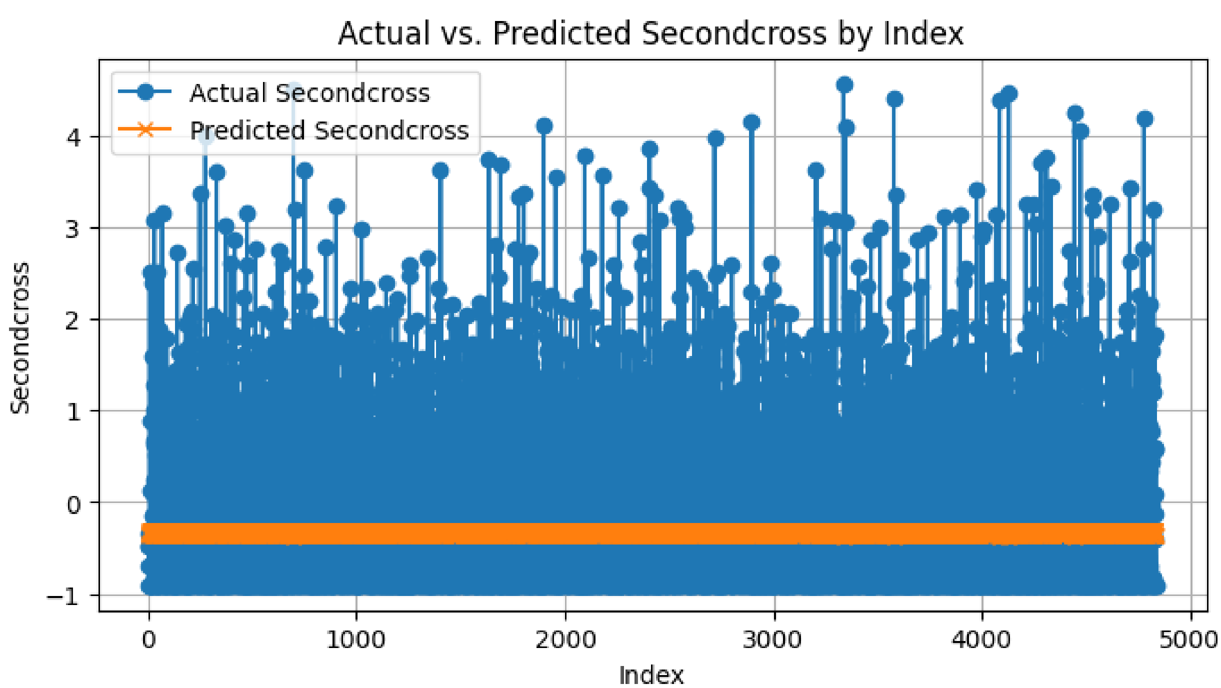


Figure 34. Actual and Predicted values Secondcross.

Figure 33 shows Actual and Predicted values of Secondcross in Munich dataset suing LSTM model. Mean absolute error (MAE) is 0.78. This is the lowest among all the models.

Table 5 below shows the comparison of all the models used in traffic flow prediction on five real time datasets.

	Datasets					
Models	Beijing	California	San Jose	Perth	Munich (1st)	Munich (2nd)
ARIMA	0.10	4406.37	9271.84	3667.27	116.07	72.68
VAR	0.094	4110.06	8167.40	2897.72	80.11	46.48
Prophet	0.10	5241.83	10252.23	3290.29	95.95	84.09
XG Boost	0.06	662.58	615.45	3114.62	11.50	7.79
LSTM	0.23	0.63	0.77	0.76	0.90	0.78

Table 5. Comparison of all the model on datasets.

The mean absolute error numbers here represent the difference between actual number of objects passing through the inductive loop detectors and other traffic data recording devices and predicted values by the models.

3.2 Image Classification

Image classification is an important step in traffic systems that can be done by using the machine learning and deep learning algorithms that help in the detection of objects. Object detection is a key part that needs to be integrated in the traffic cameras that go right with the traffic lights. For this purpose, we are going to compare two models:

i. Traditional Artificial Neural Networks (ANN's)
ii. Multi layered Convolutional Neural Networks (CNN's)

3.2.1 Traditional Artificial Neural Networks (ANN's):

(McCulloch and Pitts 1943) were the first one to have come up with extensive mathematical work on ANN's. (Grossi and Buscema 2008) describe the ANN'S as a framework in which there are nodes connected with each other internally and there are difference layers in each one of these nodes. Below is a mathematical background of ANN's:

Input: x in the equation below represents the input in the network, which is a vector. The input relates to input layer.

Forward Propagation:

Weighted Sum: The equation below represents the weighted sum which is calculated by for every input in neuron.

$$z = \Sigma(w_i * x_i) + b$$

'z' here represents the weighted sum inputs.

'w'$_I$ represents weights.

'x'$_i$ represents values for inputs.

'b' represents bias.

Activation Function: The output of a neuron is calculated by applying the activation function to the weighted sum of the inputs.

$$a = f(z)$$

'a' here is the output of neuron.

‘f’ is activation function.

Output: Output is what makes the prediction in the end.

$$y = f^{(L)} (\Sigma(w^{(L)}. a^{(L-1)}) + b^{(L)}$$

‘y’ here is the output of ANN.

‘$f^{(L)}$’ is activation function.

‘$w^{(L)}$’ are the weights.

‘$a^{(L-1)}$’ here is the previous activations vector.

‘b’ here represents bias.

3.2.2 Convolutional Neural Networks (CNN’s):

CNN has totally changed the field of image classification and has enhanced it many folds. There wasn’t one moment or study in which it can said to have been developed. But (LeChun 1990) with his seminal work changed the course of this field. His model was initially tried on handwritten digit recognition. Below is the complete mathematical breakdown of a CNN model:

Convolution Operation: this operation can be represented mathematically in following way:

$$(I * K)(x, y) = \Sigma_i \Sigma_j I(x + I, y + j) * K(I, j)$$

‘I’ represents input range.

‘K’ represents a kernel.

x and y represent spatial coordinates in the feature map.

i and j are spatial coordinated in kernel.

‘*’ is an operation.

Feature Map Dimensions: these feature maps are used to compute the dimensions after an operation of convolution.

Activation Function (ReLU): Activation function applied to the feature map element wise.

$$\text{ReLU} = \max(0, x)$$

Pooling Operation or Max-Pooling: This function takes the maximum value in a region and then down samples the map of the feature.

$$\text{Max-pooling}(x, y) = \max_{i,j} \text{FeatureMap}(x + i, j + j)$$

Fully Connected Layer: This layer connects units in all layers with each other. The input here is flattened from the feature map. This complete operation can be fully described as:

$$\text{Unit(i)} = \sum_{j=1}^{n} Weight(\text{I}, \text{j}) * \text{Input(j)} + \text{Bias(i)}$$

Loss Function (Cross-Entropy for Classification): Following function describes Loss function:

$$\text{Loss} = -\Sigma_i\, y_i \log(p(y_i))$$

Backpropagation: Backpropagation uses chain rule and calculates the gradient of the loss with respect to networks parameters. For instance, the parameter is θ the gradient here can be computed as follows:

$$\frac{\partial Loss}{\partial \theta} = \Sigma_i \frac{\partial Loss}{\partial \text{Unit(i)}} * \frac{\partial Unit(i)}{\partial \theta}$$

3.2.3 Methodology and Data:

i) Methodology:

The models will be applied to a state-of-the-art dataset. The methodology will follow in the following way:

First the Dataset will be loaded which is a built-in dataset in TensorFlow keras. Code for that will be provided in the appendix. Then data will be preprocessed. Then next step involves splitting data in train and test splits. Afterwards model will be set up and then the data processed in through the models. The last step shows the results. Below is the pictorial representation of methodology to be followed as described above:

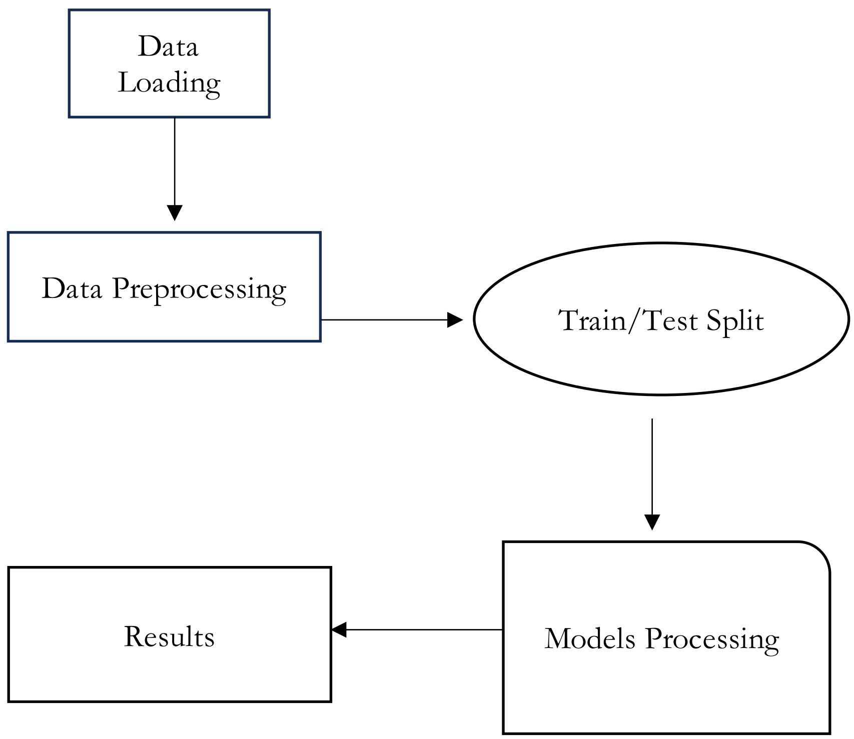


Figure 35. Methodology to be followed.

ii) Data

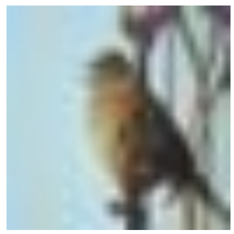
The dataset used in this experimentation is the CIFAR 10 dataset which is used for testing image classification model and detecting objects. This dataset consists of images belonging to 10 classes. Each class has 6000 images in total. Making it overall 60000 images. The images in the dataset are 32x32 size and all are color images. The training dataset is 50000 images and testing dataset is 10000 images. The classes include Automobiles, Trucks, Airplanes, Ships, Dogs, Cats, Deer, Birds, Frog and Horses. Below are the images of the classes that models are run on.

Airplane

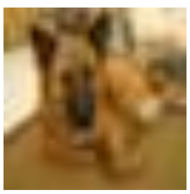
Automobile

Deer

Cat

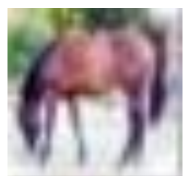


Bird

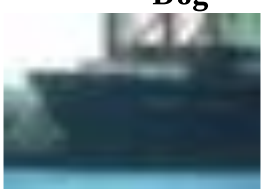


Dog

Frog



Horse



Ship

Truck

Each training batch selects images randomly. But there might be more images belonging to one class than the other one. It all level outs in the end and all the images are considered and processed in the models.

3.2.4 Experiment and Results:

The coding was done for experiments on the google colab environment. System used was MacBook Air 2017, processor 1.8 GHz Dual core Intel Core i5. Memory 8 Gb 1600 MHz DDR3. Graphics Intel HD 6000 1536 MB. TensorFlow and other libraries were downloaded by pip command all the code is attached for relevant sections in the appendix. Next is the results of the experiments. We will look at the classification reports of both the models and observe the accuracy scores separately.

Traditional ANN Experimentation and results:

After loading the data, the images were trained on a model where optimizer was set to Stochastic Gradient Descent (SGD). Loss function was sparse categorical cross entropy and number of epochs was set to 10. And the batch size was 64. For activation function first two layers of dense layer were Relu and the third layer was SoftMax. The prediction on test data set showed that the accuracy rate (which was metric for accuracy measurement) was 44%. Which is just above average. It is not a good model to be selected for object detection. Below is the classification report:

Classification Report:

	precision	recall	f1-score	support
0	0.76	0.23	0.35	1000
1	0.75	0.34	0.47	1000
2	0.35	0.48	0.40	1000
3	0.36	0.37	0.37	1000
4	0.45	0.41	0.43	1000
5	0.39	0.37	0.38	1000
6	0.56	0.52	0.54	1000
7	0.60	0.50	0.55	1000
8	0.33	0.89	0.48	1000
9	0.65	0.32	0.42	1000
accuracy			0.44	10000
macro avg	0.52	0.44	0.44	10000
weighted avg	0.52	0.44	0.44	10000

3 layered CNN experiment and results:

For this the choice of optimizer was Adam because in the traditional ANN we do not have a lot of options for the optimizer. The learning rate was set to 0.001. The loss function was categorical cross entropy. Training epoch was to 10 same as previous one and batch size to 64. There were three layers of Conv 2D and Max pooling making it 3 layered convolutional networks. The accuracy measurement metric accuracy rate shows the accuracy to be 74% for this model. Much higher than the traditional one. Below is the classification report:

Classification Report:

	precision	recall	f1-score	support
airplane	0.81	0.75	0.78	1000
automobile	0.83	0.87	0.85	1000
bird	0.70	0.62	0.66	1000
cat	0.57	0.54	0.55	1000
deer	0.67	0.70	0.68	1000
dog	0.64	0.68	0.66	1000
frog	0.83	0.76	0.79	1000
horse	0.74	0.81	0.78	1000
ship	0.81	0.86	0.84	1000
truck	0.84	0.82	0.83	1000
accuracy			0.74	10000
macro avg	0.74	0.74	0.74	10000
weighted avg	0.74	0.74	0.74	10000

The users can add more layers or reduce it and see whether the accuracy improves or not. Table 6 shows the side-by-side comparison of accuracy rate of both models.

CIFAR10 (Dataset)	
Models	Accuracy Score
Traditional ANN	44%
3 layered CNN	74%

Table 6. Comparison of image classification models for object detection

3.3 Traffic Congestion Algorithm:

This part is to recommend the most relevant traffic congestion algorithm from the existing literature because creating one from scratch is beyond the scope of this study. The algorithm mentioned in the review of relevant studies by (Bushra 2020) and colleagues. The best part of this algorithm is that it models the congestion from previous trajectories. This algorithm was used on a big dataset from Beijing which included 47 days long data of taxi trajectories. The road networks in this dataset consists of 81592 road segments. The results produced on this dataset will be good enough to fit into our system.

3.4 Intelligent Incident Detection:

This part is also a recommendation for selecting a model that is relevant for our system. This will play an important role in the detection of incidents. In the above discussion it has been reviewed already what kind of incidents are included under this umbrella. A comprehensive review done by (Samia and Abdeslem 2020) on the topic of intelligent incident detection. Which included most important studies in this field and then compared the models and techniques used in them. This showed that research by (Yang 2009) and colleagues gives the highest accuracy in the detection of incidents in urbanized road network settings.

The model used here was a combination of Support Vector Machine (SVM) and Fuzzy Logic (FL). This experiment was done by the authors on PARAMIC traffic simulator. The accuracy measures used for this were incident detection rate (DR), False Alarm Rate (FAR) and Mean Time to Detect (MTTD). DR was 95.7%, FAR was 1.6% and MTTD was 29 seconds. These models can be used in our ITMS for the better detection of incidents.

CHAPTER Four

Database Design

This chapter gives recommendation for the design of database and how the traffic data is to be recorded what are the techniques and software's that can be deployed for the recording of data. Keep track and change the relationship between different aspects of traffic as we shall see ahead. Here we will also briefly look at the Graphical User Interface (GUI). GUI design is not part of this project but in overall design it important to keep in check of every detail.

4.1 Database Design

4.1.1 Data Schema

The data schema for the ITMS should include tables and fields for storing data related to traffic cameras, sensors, vehicle detections, traffic flow predictions, and other relevant information. It should be designed to support efficient querying and analysis of traffic data, as well as the storage and retrieval of AI model outputs.

4.1.2 Tables and Relationships

Incidents: Stores information about traffic incidents, such as accidents, road closures, or construction, including the location, type, severity, and duration of the incident.
Alerts: Stores data about generated alerts and notifications sent to traffic authorities, including the timestamp, alert type, target recipient, and associated incident or traffic situation.

Traffic Patterns: Stores historical traffic pattern data, such as average traffic volumes, speeds, and occupancy rates for different times of the day, days of the week, and seasons.
These tables should have appropriate relationships, such as foreign key, to establish connections between different types of data. For example:

Vehicle Detections table could have a foreign key to the Cameras table, linking each detection to the camera that captured it.

Traffic Flow Predictions table could have a foreign key to the Sensors table, linking each prediction to the sensor from which the data was collected.

Alerts table could have a foreign key to the Incidents table, linking each alert to the corresponding incident or traffic situation.

4.1.3 Database Management System (DBMS)

The ITMS should use a robust and scalable DBMS to manage its data storage and retrieval needs. The chosen DBMS should support the necessary data schema and relationships, as well as provide efficient querying, indexing, and data analysis capabilities. Some popular DBMS options for this purpose include:

• PostgreSQL: An open-source, object-relational DBMS known for its performance, extensibility, and support for spatial data types, making it a suitable choice for managing geospatial data related to traffic cameras and sensors.

• MySQL: Another open-source, relational DBMS that offers a wide range of features and performance optimizations, suitable for handling large-scale traffic data.

• Microsoft SQL Server: A commercial, relational DBMS that provides a comprehensive set of features and tools for data management, analysis, and security. It also offers built-in support for spatial data types and geospatial queries.

When selecting a DBMS, consider factors such as the expected scale of the traffic data, the complexity of the data schema and relationships, and the requirements for data analysis and reporting.

4.2 Graphical User Interface (GUI)

This part of the chapter will at the GUI. Goal here is to look at the techniques that can be used to layout design and set up GUIs for the ITMS. This must be user friendly because those people who are going to interact with this on a regular basis are not going to be experts.

4.2.1 Layout and Design

The GUI for the ITMS should be designed with usability, functionality, and responsiveness in mind. It should provide an intuitive and easy-to-navigate interface that allows traffic authorities and other stakeholders to access relevant traffic data, AI-generated insights, and management tools. The layout should be organized into logical sections, with clear labels and icons for different features and functions. Consider using a modern UI framework, such as Bootstrap or Material Design, to create a visually appealing and user-friendly interface.

4.2.2 Functionality

Key functionalities of the GUI include:

- Real-time traffic monitoring: Display live traffic camera feeds, sensor data, and traffic flow visualizations on a map.
- Traffic data analysis: Access historical traffic data, AI-generated traffic predictions, and performance metrics.
- Incident management: View, add, or edit traffic incidents, and send alerts to relevant authorities.
- Traffic management tools: Adjust traffic signal timings, lane assignments, and rerouting suggestions based on AI-generated recommendations.

4.2.3 Mockups or Wireframes

To help visualize the design and layout of the GUI, create mockups or wireframes that illustrate the main interface components and their organization. These visual aids will serve as a blueprint for developers during the implementation phase and help ensure a coherent and consistent user experience. Use tools like Sketch, Adobe XD, or Figma to create high-fidelity mockups or wireframes that can be easily shared and reviewed with the project team.

CHAPTER Five

Hardware and Software components

The goal in this chapter is to discuss the present the best ideas forward for the hardware and software components that will be going into the ITMS. This is the final part that we need to look at to make a deployable ITMS. First, we will look at the implementation strategy. Software development process and project management techniques. Hardware components will then follow. It is important in the end to provide an idea for the users to test and validate the models and techniques that are to be deployed.

5.1 Implementation Strategy

5.1.1 System Architecture

Develop a system architecture that outlines the main components of the ITMS, their interactions, and communication protocols. This may include a diagram showing the relationship between hardware components (e.g., cameras, sensors, central server), software components (e.g., traffic monitoring application, data processing and storage, AI model training and deployment), and external systems (e.g., traffic authorities, existing traffic infrastructure).

5.1.2 Software Development

Adopt a software development methodology, such as Agile or Scrum, to manage the development process. This approach will help ensure efficient collaboration, timely delivery, and iterative improvement of the software components. Break down the project into manageable tasks, and assign them to developers with relevant expertise (e.g., AI modeling, database design, frontend development). Use version control systems, like Git, to track code changes and collaborate effectively.

5.1.3 Hardware Integration

Develop a plan for procuring, installing, and maintaining the necessary hardware components, such as traffic cameras, sensors, and the central server. This should include selecting appropriate hardware specifications, identifying suitable installation locations, and establishing maintenance schedules. Ensure proper integration with the software components of the ITMS and test the system thoroughly to validate its performance and reliability.

5.1.4 Communication Between Components

Design and implement communication protocols between the various components of the ITMS. This may involve:

- Defining data formats and APIs for exchanging information between cameras, sensors, and the central server.
- Implementing secure and reliable data transmission methods, such as encrypted connections or message queues.
- Developing middleware to manage the communication between the software components, AI models, and external systems.
- Ensure that the communication protocols are well-documented, robust, and scalable to accommodate the growth and evolution of the ITMS.

5.2 Testing and Validation

5.2.1 Test Scenarios and Methods

Develop comprehensive test scenarios and methods to ensure the quality and performance of the ITMS. This may include:

- Unit testing: Verify the functionality of individual software components and algorithms.

- Integration testing: Test the interaction and compatibility of different components within the system.
- System testing: Evaluate the overall performance of the ITMS, including hardware and software components, under various traffic conditions and use cases.
- User acceptance testing: Obtain feedback from traffic authorities and other stakeholders on the usability and effectiveness of the system.

5.2.2 Validation Criteria

Establish validation criteria for the ITMS to measure its performance and success. This may include metrics such as:

- Detection accuracy: The percentage of vehicles correctly identified by the AI models.
- Prediction accuracy: The deviation between the AI-generated traffic flow predictions and the actual traffic data.
- Response time: The time taken for the system to generate alerts and recommendations in response to changing traffic conditions.
- User satisfaction: Feedback from traffic authorities and other stakeholders on the usability and effectiveness of the system.

5.2.3 Security and Load Testing

Perform security and load testing to ensure the resilience and stability of the ITMS under various conditions. This may involve:

- Security testing: Identify and mitigate potential vulnerabilities in the system, such as unauthorized access or data breaches.
- Load testing: Assess the system's ability to handle high volumes of traffic data, concurrent users, and other performance demands.

5.3 Deployment and Monitoring

5.3.1 Deployment Process

Develop a deployment process for the ITMS, which includes:

- Installing and configuring hardware components, such as cameras, sensors, and the central server.
- Deploying the software components, including the traffic monitoring application, data processing and storage systems, and AI models.
- Integrating the ITMS with existing traffic infrastructure and authorities.

5.3.2 Training for Traffic Authorities

Provide training and support for traffic authorities and other stakeholders on using the ITMS. This may involve:

- Conducting workshops or training sessions to familiarize users with the system's features and functionalities.
- Providing documentation and user guides that explain how to use the system effectively.
- Offering ongoing support and assistance to address any issues or questions that may arise during the system's operation.

5.3.3 Continuous Monitoring and Improvement

Monitor the performance of the ITMS regularly and make improvements based on user feedback, performance metrics, and technological advancements. This may involve:

- Collecting and analyzing system performance data to identify areas for improvement.
- Implementing updates and enhancements to the software components, AI models, and hardware infrastructure.

- Engaging with traffic authorities and other stakeholders to gather feedback and suggestions for future development.

5.4 Proposed Framework

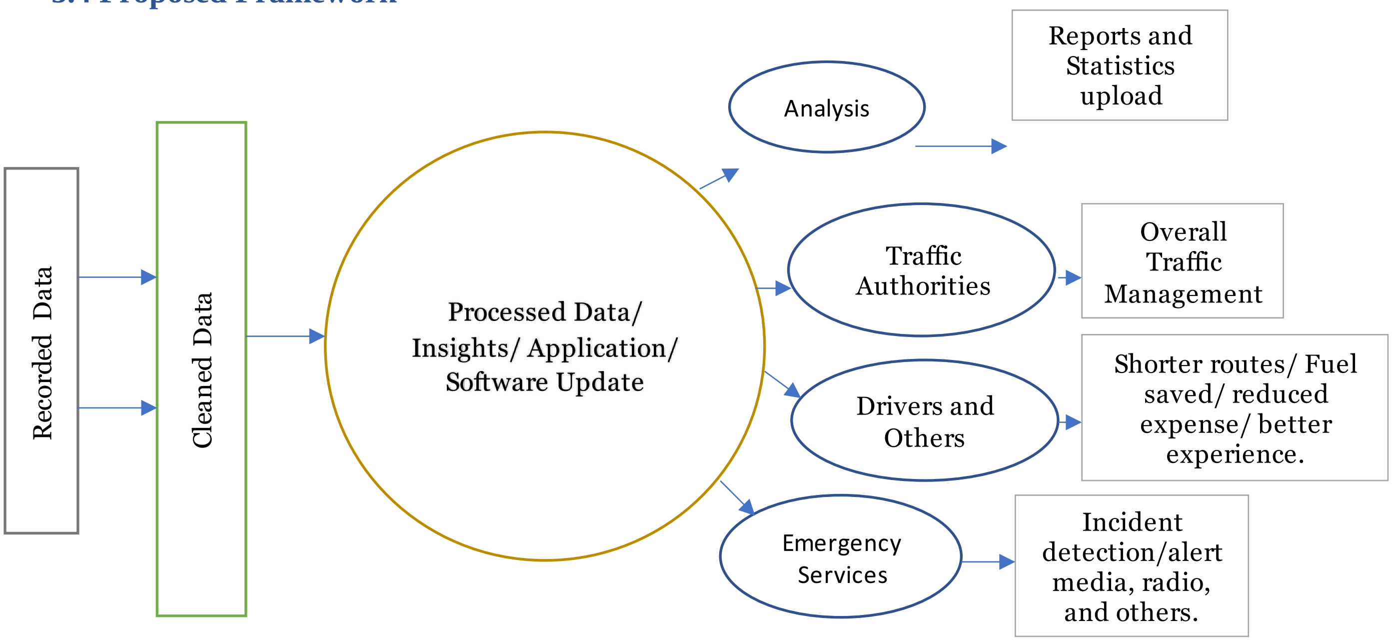


Figure 36. Proposed Framework

Figure 36 above shows the proposed framework that can be taken out of this whole comprehensive analysis of Intelligent Traffic Management Systems (ITMS).

Chapter Six

Conclusion and Future Development

6.1 Conclusion

In this project we have discussed extensively key components of ITMS. The reason this is an important field that demands a lot of research attention is because it is a key part and plays a critical role in the future smart cities and urban planning. Most of the research done in it not only concerns future but the present accurate prediction of traffic flow and reduced congestion which is critical in the flow of communication through transportation. This predictive analysis saves a lot of us from paying higher economic cost which is estimated to be at $88 billion is USA, per driver around $1377. All these reasons demanded us to answer a key question. Is there any one way we can predict traffic and reduce the congestion in longer term which is so vital for overall transportation?

A comprehensive literature review shows us that there isn't one size fit all model or one way which can help us achieve our goals here. Although the models that used both spatial and temporal data built on deep learning models outperformed baseline statistical models (SM) and machine learning (ML) models. Our own comparison shows us that LSTM outperformed other SM and ML models by giving higher accuracy on four out of five real time datasets. The literature reviewed showed us that expanding mass transit can do a better job in tackling prediction issues by extension congestion issues. Because after a point enough cars and trucks that take up a lot of space will cause congestion. Even the road expansion showed to lead in the increase of traffic in the shorter and longer run by 3% to 6% in the former and up to 10% in the latter.

For image classification we compared ANN and a 3 layered CNN model on a CIFAR dataset that had ten different classes of objects for classification. Results showed that 3 layered CNN model gave higher accuracy. It led us to the next part of incident detection where literature has classified two types of incidents first one being planned and second one being unplanned. Both the types of incident detection can be done using a SVM model combined with a fuzzy logic algorithm. As this model showed to outperform others in the existing research. For the next part which is algorithmic way of predicting traffic congestion. Change in speed method does a better job than cut of speed according to the existing research. And it is better to be used in future works as well.

Apart from these techniques that directly focus on traffic reduction and better management of traffic. The database and traffic recording techniques are also discussed here. Most of the traffic prediction now happens in real time especially in the big cities. Efficient transportation and traffic management requires big data to be live streamed now due to the pressure of traffic overall. The GUI part and the key functionalities of the interface are also overviewed.

In the end it matters a great deal that how cost efficient all the ITMS set up is. Because relevant authorities must be able to integrate it with the existing traffic infrastructure. Economic factors are a big reason why most of the advanced traffic management set ups are seen only in the big cities across the world. We discuss how to test and validate software's and hardware components, develop a deployment process, and training of traffic authorities. These all tools and techniques will help us in Intelligently managing traffic for better communication and lead to better urban management.

6.2 Recommendations for Future Development

As technology and AI capabilities continue to evolve, there are numerous opportunities for future development and enhancements to the ITMS. Some recommendations include: Expanding the scope of the system to include other modes of transportation, such as public transit, bicycles, and pedestrians.

Incorporating additional data sources, such as social media or mobile app data, to improve traffic predictions and analysis. Developing advanced AI models that can adapt to changing traffic conditions in real time. Integrating with emerging transportation technologies, such as connected vehicles, autonomous vehicles, and smart infrastructure, to further optimize traffic flow and safety. Implementing advanced traffic management strategies, such as dynamic pricing for toll roads or congestion charges, to incentivize more efficient use of transportation resources.

Enhancing the user interface and visualization capabilities of the system to provide more insightful and actionable information for traffic authorities and other stakeholders. Exploring partnerships with private companies, such as ride-sharing services and navigation apps, to leverage their data and resources for more effective traffic management. Assessing the environmental impact of the ITMS and developing strategies to reduce emissions and promote sustainable transportation solutions. Collaborating with academic institutions and research organizations to stay updated on the latest advancements in AI, transportation, and urban planning, and to continuously improve the ITMS.

By continuously evolving and adapting to new technologies and challenges, the ITMS has the potential to become an invaluable tool for traffic management and a model for innovative, data- driven transportation solutions.

References

A fuzzy logic model for real-time incident detection in urban road network. (2013). Proceedings of the 5th International Conference on Agents and Artificial Intelligence. https://doi.org/10.5220/0004239904650472

Ahmed, F., & Hawas, Y. E. (2012). A threshold-based real-time incident detection system for Urban Traffic Networks. Procedia - Social and Behavioral Sciences, 48, 1713–1722. https://doi.org/10.1016/j.sbspro.2012.06.1146

Ahn, J. Y., Ko, E., & Kim, E. (2015). Predicting spatiotemporal traffic flow based on support vector regression and bayesian classifier. *2015 IEEE Fifth International Conference on Big Data and Cloud Computing*. https://doi.org/10.1109/bdcloud.2015.64

Aköz, Ö., & Karsligil, M. E. (2011). Traffic event classification at intersections based on the severity of abnormality. Machine Vision and Applications, 25(3), 613–632. https://doi.org/10.1007/s00138-011-0390-4

Al-Selwi, H. F., Bin Abd Aziz, A., Abas, F. S., Amir Hamzah, N. A., & Mahmud, A. B. (2022). The impact of weather data on traffic flow prediction models. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *11*(4), 1223. https://doi.org/10.11591/ijai.v11.i4.pp1223-1231

Albertengo, G., & Hassan, W. (2018). Short term urban traffic forecasting using Deep Learning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-4/W7*, 3–10. https://doi.org/10.5194/isprs-annals-iv-4-w7-3-2018

Altundogan, T. G., & Karakose, M. (2019). Image processing and deep neural image classification based physical feature determiner for traffic stakeholders. 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG). https://doi.org/10.1109/sgcf.2019.8782371

Analysis of freeway traffic time-series data by using box ... (n.d.-a). https://onlinepubs.trb.org/Onlinepubs/trr/1979/722/722-001.pdf

Anthopoulos, L., Janssen, M., & Weerakkody, V. (2019). A unified smart city model (USCM) for smart city conceptualization and benchmarking. Smart Cities and Smart Spaces, 247–264. https://doi.org/10.4018/978-1-5225-7030-1.ch011

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the Arima model. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. https://doi.org/10.1109/uksim.2014.67

Average daily traffic. San Jose CA Open Data Portal. (2023, October 21). https://data.sanjoseca.gov/dataset/average-daily-traffic

Axenie, C., & Bortoli, S. (2020, February 8). Road traffic prediction dataset. Zenodo. https://zenodo.org/records/3653880

Böther, M., Schiller, L., Fischbeck, P., Molitor, L., Krejca, M. S., & Friedrich, T. (2021). Evolutionary minimization of traffic congestion. Proceedings of the Genetic and Evolutionary Computation Conference. https://doi.org/10.1145/3449639.3459307

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative K-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, *62*, 21–34. https://doi.org/10.1016/j.trc.2015.11.002

Cars still dominate the American commute. World Economic Forum. (n.d.). https://www.weforum.org/agenda/2022/05/commute-america-sustainability-cars/

Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications*, *36*(3), 6164–6173. https://doi.org/10.1016/j.eswa.2008.07.069

Chai, D., Wang, L., & Yang, Q. (2018). Bike flow prediction with multi-graph convolutional networks. Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. https://doi.org/10.1145/3274895.3274896

Chen, T., & Guestrin, C. (2016). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2939672.2939785

Chlyah, M., Dardor, M., & Boumhidi, J. (2016). Multi-agent system based on support vector machine for incident detection in Urban Roads. 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA). https://doi.org/10.1109/sita.2016.7772285

CIFAR-10 and CIFAR-100 datasets. (n.d.). https://www.cs.toronto.edu/~kriz/cifar.html

Clark, S. (2003). Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering*, *129*(2), 161–168. https://doi.org/10.1061/(asce)0733-947x(2003)129:2(161)

Clark, S. D., Dougherty, M. S., & Kirby, H. R. (1992, November 30). *The use of neural networks and time series models for short term Traffic forecasting: A comparative study*. TRID. https://trid.trb.org/view/390622

Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, *118*, 102674. https://doi.org/10.1016/j.trc.2020.102674

Dadi, H. S., & Mohan Pillutla, G. K. (2016). Improved face recognition rate using hog features and SVM classifier. IOSR Journal of Electronics and Communication Engineering, 11(04), 34–44. https://doi.org/10.9790/2834-1104013444

Dardor, M., Chlyah, M., & Boumhidi, J. (2018). Incident detection in signalized urban roads based on genetic algorithm and Support Vector Machine. 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). https://doi.org/10.1109/isacv.2018.8354029

de Medrano, R., & Aznarte, J. L. (2020). A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction. Applied Soft Computing, 96, 106615. https://doi.org/10.1016/j.asoc.2020.106615

Dehuai Zeng, Jianmin Xu, Jianwei Gu, Liyan Liu, & Gang Xu. (2008). Short term traffic flow prediction based on online learning SVR. *2008 Workshop on Power Electronics and Intelligent Transportation System*. https://doi.org/10.1109/peits.2008.134

Dia, H. (1998, November 30). *Freeway travel time estimation using Neural Networks*. TRID. https://trid.trb.org/View/1164447

Dia, Hussein. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, *131*(2), 253–261. https://doi.org/10.1016/s0377-2217(00)00125-9

Dong, H. (2022). Prediction Model of Urban Short-time Traffic Flow Based on Intelligent Algorithm. *IEEE*.

Dougherty, M. S., & Cobbett, M. R. (1997). Short-term Inter-urban traffic forecasts using Neural Networks. *International Journal of Forecasting*, *13*(1), 21–31. https://doi.org/10.1016/s0169-2070(96)00697-8

Elsevier: An information analytics business. www.elsevier.com. (n.d.). https://www.elsevier.com/

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627–1645. https://doi.org/10.1109/tpami.2009.167

Feng, X., Ling, X., Zheng, H., Chen, Z., & Xu, Y. (2019). Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction. *IEEE Transactions on*

Intelligent Transportation Systems, *20*(6), 2001–2013. https://doi.org/10.1109/tits.2018.2854913

Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Transactions on Intelligent Transportation Systems, 10(2), 246–254. https://doi.org/10.1109/tits.2009.2021448

Ghosh, Bidisha, & Smith, D. P. (2014). Customization of automatic incident detection algorithms for signalized urban arterials. Journal of Intelligent Transportation Systems, 18(4), 426–441. https://doi.org/10.1080/15472450.2013.806843

GitHub. (n.d.). https://github.com/codebasics/deep-learning-keras-tf tutorial/blob/master/16_cnn_cifar10_small_image_classification/cnn_cifar10_dataset.ipynb

Google. (n.d.). Google scholar. https://scholar.google.com/scholar?hl=en&q=Tang%2C%2BY.%2C%2BLiu%2C%2BW.%2C%2BSun%2C%2BL.%2C%2BWei%2C%2BF.%2Band%2BYu%2C%2BC.%2B2015.%2BApplication%2Bof%2Bimproved%2Btime%2Bseries%2Bmodel%2Bin%2Bforecasting%2Bof%2Bshort-term%2Btraffic%2Bflow%2Bfor%2Bfreeway.%2BApplication%2BResearch%2Bof%2BComputers%2C%2B32%281%29%3A146-149.&inst=3396366540409011505

Google. (n.d.). Google search. https://www.google.com/search?q=User%2BGuide%2Bof%2BT-Drive%2BData%2BVersion%2B1.0%2BUpdated%2Bon%2BAugust%2B12%2C%2B2011&oq=User%2BGuide%2Bof%2BT-

Drive%2BData%2BVersion%2B1.0%2BUpdated%2Bon%2BAugust%2B12%2C%2B2011&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTI1MTJqMGoxNagCALACAA&sourceid=chrome&ie=UTF-8

Grossi, E., & Buscema, M. (2007). Introduction to artificial neural networks. European Journal of Gastroenterology & Hepatology, 19(12), 1046–1054. https://doi.org/10.1097/meg.0b013e3282f198a0

Gu, Y., Qian, Z. (Sean), & Chen, F. (2016). From Twitter to detector: Real-time traffic incident detection using social media data. Transportation Research Part C: Emerging Technologies, 67, 321–342. https://doi.org/10.1016/j.trc.2016.02.011

Guberfain Do Amaral, B., Nasser, R., Casanova, M. A., & Lopes, H. (2016). Busesinrio: Buses as mobile traffic sensors: Managing the bus GPS data in the City of Rio de Janeiro. 2016 17th IEEE International Conference on Mobile Data Management (MDM). https://doi.org/10.1109/mdm.2016.66

Guo, G., Yuan, W., Liu, J., Lv, Y., & Liu, W. (2023). Traffic forecasting via dilated temporal convolution with peak-sensitive loss. *IEEE Intelligent Transportation Systems Magazine*, *15*(1), 48–57. https://doi.org/10.1109/mits.2021.3119869

Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 922–929. https://doi.org/10.1609/aaai.v33i01.3301922

Hamed, M. M., Al-Masaeid, H. R., & Said, Z. M. (1995). Short-term prediction of traffic volume in urban arterials. Journal of Transportation Engineering, 121(3), 249–254. https://doi.org/10.1061/(asce)0733-947x(1995)121:3(249)

Han, D., Chen, J., & Sun, J. (2019). A parallel spatiotemporal deep learning network for Highway Traffic Flow Forecasting. *International Journal of Distributed Sensor Networks*, *15*(2), 155014771983279. https://doi.org/10.1177/1550147719832792

Handy, S. (2020, September 18). *Increasing highway capacity unlikely to relieve traffic congestion*. eScholarship, University of California. https://escholarship.org/uc/item/58x8436d

Hatri, C. E., & Boumhidi, J. (2017). Fuzzy deep learning based urban traffic incident detection. 2017 Intelligent Systems and Computer Vision (ISCV). https://doi.org/10.1109/isacv.2017.8054903

Hawas, Y. E., & Ahmed, F. (2016). A binary logit-based incident detection model for Urban Traffic Networks. Transportation Letters, 9(1), 49–62. https://doi.org/10.1080/19427867.2016.1157132

He, S., Luo, Q., Du, R.-H., Fu, H., Li, H., & He, G. (2023). *Stgc-GNNS: A GNN-Based Traffic Prediction Framework with a Spatial-Temporal Granger Causality Graph*. https://doi.org/10.2139/ssrn.4378318

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Högberg, P. (1976). Estimation of parameters in models for traffic prediction: A non-linear regression approach. *Transportation Research*, *10*(4), 263–265. https://doi.org/10.1016/0041-1647(76)90059-9

Hong, H., Huang, W., Xing, X., Zhou, X., Lu, H., Bian, K., & Xie, K. (2015). Hybrid multi-metric K-nearest neighbor regression for traffic flow prediction. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. https://doi.org/10.1109/itsc.2015.365

Hong, W.-C., Dong, Y., Zheng, F., & Lai, C.-Y. (2011). Forecasting urban traffic flow by SVR with continuous ACO. *Applied Mathematical Modelling*, *35*(3), 1282–1291. https://doi.org/10.1016/j.apm.2010.09.005

Hooda, W., Yadav, P. K., Bhole, A., & Chaudhari, D. D. (2016). An image processing approach to Intelligent Traffic Management System. Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. https://doi.org/10.1145/2905055.2905091

Hossain, B., Adnan, K. A., Rabbi, M. F., & Ali, M. E. (2020). Modelling road traffic congestion from trajectories. Proceedings of the 3rd International Conference on Data Science and Information Technology. https://doi.org/10.1145/3414274.3414491

Hosseini, S. H., Moshiri, B., Rahimi-Kian, A., & Araabi, B. N. (2012). Short-term traffic flow forecasting by mutual information and Artificial Neural Networks. *2012 IEEE International Conference on Industrial Technology*. https://doi.org/10.1109/icit.2012.6210093

Hu, J., Gao, P., Yao, Y., & Xie, X. (2014). Traffic flow forecasting with particle swarm optimization and support vector regression. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). https://doi.org/10.1109/itsc.2014.6958049

Ijitee. (n.d.-b). https://www.ijitee.org/wp-content/uploads/papers/v8i3c/C10680183C19.pdf

Jain, V., Sharma, A., & Subramanian, L. (2012). Road traffic congestion in the developing world. Proceedings of the 2nd ACM Symposium on Computing for Development. https://doi.org/10.1145/2160601.2160616

Jenkins, G. M., & P., B. G. E. (1976). Time Series Analysis Forecasting and control: Rev. ed. Holden-Day.

Jeong, Y.-S., Byon, Y.-J., Castro-Neto, M. M., & Easa, S. M. (2013). Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 14(4), 1700–1707. https://doi.org/10.1109/tits.2013.2267735

Jiang, T., & Zhang, Z. (2022). Kalman filter using SOV model with maximum versoria criterion for short-term traffic flow forecasting. *Proceedings of the 6th International Conference on Digital Signal Processing*. https://doi.org/10.1145/3529570.3529579

Khekare, G. S., & Sakhare, A. V. (2013). A smart city framework for intelligent traffic system using VANET. 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s). https://doi.org/10.1109/imac4s.2013.6526427

Kumar, S. V., & Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, *7*(3). https://doi.org/10.1007/s12544-015-0170-8

Kyle Funk, Lindsey Volz. (2022, August 3). Innovative ways to deal with Traffic Congestion & Road Funding. National League of Cities. https://www.nlc.org/article/2022/08/03/innovative-ways-to-deal-with-traffic-congestion-road-funding/

Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., & Hubbard, W. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE Communications Magazine, 27(11), 41–46. https://doi.org/10.1109/35.41400

Lee, E.-M., Kim, J.-H., & Yoon, W.-S. (n.d.). Traffic speed prediction under weekday, time, and neighboring links' speed: Back propagation neural network approach. *Lecture Notes in Computer Science*, 626–635. https://doi.org/10.1007/978-3-540-74171-8_62

Lee, S., & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, *1678*(1), 179–188. https://doi.org/10.3141/1678-22

Leventis, D. (2022, January 2). XGBoost mathematics explained. Medium. https://dimleve.medium.com/xgboost-mathematics-explained-58262530904a

Li, R., Jiang, C., Zhu, F., & Chen, X. (2016). Traffic flow data forecasting based on interval type-2 fuzzy sets theory. *IEEE/CAA Journal of Automatica Sinica*, *3*(2), 141–148. https://doi.org/10.1109/jas.2016.7451101

Li, S., Shen, Z., & Wang, F.-Y. (2012). A weighted pattern recognition algorithm for short-term traffic flow forecasting. *Proceedings of 2012 9th IEEE International Conference on Networking, Sensing and Control*. https://doi.org/10.1109/icnsc.2012.6204881

Li, S., Shen, Z., & Xiong, G. (2012). A K-nearest neighbor locally weighted regression method for short-term traffic flow forecasting. *2012 15th International IEEE Conference on Intelligent Transportation Systems*. https://doi.org/10.1109/itsc.2012.6338648

Liang, Y., Jiang, Z., & Zheng, Y. (2017). Inferring traffic cascading patterns. Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. https://doi.org/10.1145/3139958.3139960

Liu, H., Zhang, L., & Xin, S. (2021). An improved target detection general framework based on yolov4. 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO). https://doi.org/10.1109/robio54168.2021.9739530

Liu, J., & Guan, W. (n.d.). A summary of traffic flow forecasting methods | semantic scholar. https://www.semanticscholar.org/paper/A-Summary-of-Traffic-Flow-Forecasting-Methods-Wei/4cf1941d3cecb6f9b28795848ee17b1cfce33e99

Liu, T., Jiang, A., Zhou, J., Li, M., & Kwan, H. K. (2023). GraphSAGE-based dynamic spatial–temporal graph convolutional network for traffic prediction. *IEEE Transactions on*

Intelligent Transportation Systems, *24*(10), 11210–11224. https://doi.org/10.1109/tits.2023.3279929

Liu, W., Zheng, Y., Chawla, S., Yuan, J., & Xing, X. (2011). Discovering spatio-temporal causal interactions in traffic data streams. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2020408.2020571

Long, J., Gao, Z., Ren, H., & Lian, A. (2008). Urban traffic congestion propagation and Bottleneck Identification. Science in China Series F: Information Sciences, 51(7), 948–964. https://doi.org/10.1007/s11432-008-0038-9

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic flow prediction with Big Data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 1–9. https://doi.org/10.1109/tits.2014.2345663

Lv, Y., Tang, S., & Zhao, H. (2009). Real-time highway traffic accident prediction based on the K-nearest neighbor method. *2009 International Conference on Measuring Technology and Mechatronics Automation*. https://doi.org/10.1109/icmtma.2009.657

Madhu Sanjeevi (Mady. (2018, January 21). Chapter 10.1: Deepnlp - LSTM (long short term memory) networks with math. Medium. https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

Mandayam, C. V., & Prabhakar, B. (2014). Traffic congestion. The 2014 ACM International Conference on Measurement and Modeling of Computer Systems. https://doi.org/10.1145/2591971.2592014

Mankinen, T., Stoklasa, J., & Luukka, P. (2022). Predicting short-term traffic speed and speed drops in the urban area of a medium-sized European city—a traffic control and decision support perspective. *Intelligent Systems and Applications in Business and Finance*, 163–187. https://doi.org/10.1007/978-3-030-93699-0_7

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115–133. https://doi.org/10.1007/bf02478259

Nguyen, H., Liu, W., & Chen, F. (2017). Discovering congestion propagation patterns in spatio-temporal traffic data. IEEE Transactions on Big Data, 3(2), 169–180. https://doi.org/10.1109/tbdata.2016.2587669

Nikolaev, A. B., & Sapego, Y. S. (2017). Implementation of incident detection algorithm based on Fuzzy Logic in PTV vissim. International Journal of Advanced Studies, 6(4), 37. https://doi.org/10.12731/2227-930x-2016-4-37-45

Nisha, & Kumar, K. (2022). Multi-branch traffic flow prediction based on temporal speed. *Proceedings of the Sixth International Conference of Transportation Research Group of India*, 47–61. https://doi.org/10.1007/978-981-19-3494-0_4

Oh, S., Ritchie, S. G., & Oh, C. (2002). Real-time traffic measurement from single loop inductive signatures. *Transportation Research Record: Journal of the Transportation Research Board*, *1804*(1), 98–106. https://doi.org/10.3141/1804-14

Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological*, *18*(1), 1–11. https://doi.org/10.1016/0191-2615(84)90002-x

Pan, B., & Wu, H. (2017). Urban traffic incident detection with mobile sensors based on SVM. 2017 XXXIInd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS). https://doi.org/10.23919/ursigass.2017.8104994

Papageorgiou, C. P., Oren, M., & Poggio, T. (n.d.). A general framework for object detection. Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). https://doi.org/10.1109/iccv.1998.710772

Papers | semantic scholar. (n.d.-c). https://www.semanticscholar.org/paper/

Park, J., Li, D., Murphey, Y. L., Kristinsson, J., McGee, R., Kuang, M., & Phillips, T. (2011). Real time vehicle speed prediction using a neural network traffic model. *The 2011 International Joint Conference on Neural Networks*. https://doi.org/10.1109/ijcnn.2011.6033614

Parkany, A. E. (2005). In A complete review of Incident Detection Algorithms & their deployment: What works and what doesn't. Storrs, CT; New England Transportation Consortium.

Pavlyuk, D. (2022). Robust and responsive learning of spatiotemporal urban traffic flow relationships. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 14524–14541. https://doi.org/10.1109/tits.2021.3130146

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *79*, 1–17. https://doi.org/10.1016/j.trc.2017.02.024

Qiu, C., Wang, C., Zuo, X., & Fang, B. (2011). A bayesian regularized neural network approach to short-term traffic speed prediction. *2011 IEEE International Conference on Systems, Man, and Cybernetics*. https://doi.org/10.1109/icsmc.2011.6084006

Qureshi, K. N., & Abdullah, A. H. (2013). A Survey on Intelligent Transportation Systems. Johor Darul Takzim.

Runkle, D. E. (1987). Vector Autoregressions and Reality. https://doi.org/10.21034/sr.107

Samia, H., & Abdeslem, D. (2020). A review of artificial intelligence techniques used for Urban Automatic Incident Detection Systems. Proceedings of the 2020 9th International Conference on Software and Computer Applications. https://doi.org/10.1145/3384544.3384557

Schimbinschi, F., Moreira-Matias, L., Nguyen, V. X., & Bailey, J. (2017). Topology-regularized universal vector autoregression for traffic forecasting in large urban areas. *Expert Systems with Applications*, *82*, 301–316. https://doi.org/10.1016/j.eswa.2017.04.015

Schuszter, I. C. (2017). A comparative study of machine learning methods for Traffic sign recognition. 2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). https://doi.org/10.1109/synasc.2017.00070

Senthilnayaki, B., Rajeswary, C., Nivetha, G., Dharanyadevi, P., Mahalakshmi, G., & Devi, A. (2022). Traffic sign prediction and classification using Image Processing Techniques. 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN). https://doi.org/10.1109/icstsn53084.2022.9761286

Slideshare. (2014, November 1). *Capitulo 8*. SlideShare. https://www.slideshare.net/straingly/32784

Smith, B L, & Demetsky, M. J. (1993, November 30). *Short-term traffic flow prediction: Neural Network Approach*. Transportation Research Record. https://trid.trb.org/View/424677

Smith, Brian L., & Demetsky, M. J. (1996). Multiple-interval freeway traffic flow forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, *1554*(1), 136–141. https://doi.org/10.1177/0361198196155400117

Stock, J. H., & Watson, M. W. (n.d.). Vector Autoregressions. Journal of Economic Perspectives. https://www.aeaweb.org/articles?id=10.1257%2Fjep.15.4.101#:~:text=Two%20decades%20ago%2C%20Christopher%20Sims,by%20its%20own%20lagged%20values.

Taehyung Kim, Hyoungsoo Kim, & Lovell, D. J. (n.d.). Traffic flow forecasting: Overcoming memoryless property in nearest neighbor non-parametric regression. *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005*. https://doi.org/10.1109/itsc.2005.1520181

Takumi, K., Watanabe, K., Ha, Q., Tejero-De-Pablos, A., Ushiku, Y., & Harada, T. (2017). Multispectral object detection for Autonomous Vehicles. Proceedings of the on Thematic Workshops of ACM Multimedia 2017. https://doi.org/10.1145/3126686.3126727

Tang, W. M., Yiu, K. F. C., Chan, K. Y., & Zhang, K. (2023). Conjoining congestion speed-cycle patterns and deep learning neural network for short-term traffic speed forecasting. *Applied Soft Computing*, *138*, 110154. https://doi.org/10.1016/j.asoc.2023.110154

Taylor, S. J., & Letham, B. (2017). Forecasting at Scale. https://doi.org/10.7287/peerj.preprints.3190v1

Tian, J., Zou, F., Guo, F., Gu, Q., Ren, Q., & Xu, G. (2021). Expressway traffic flow forecasting based on SF-RF model via ETC data. *International Conference on Frontiers of Electronics, Information and Computation Technologies*. https://doi.org/10.1145/3474198.3478238

Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. https://doi.org/10.1109/smartcity.2015.63

Ting, T. J., Sanner, S., & Abdulhai, B. (2023). A comparative evaluation of established and Contemporary Deep Learning Traffic Prediction Methods. *Handbook on Artificial Intelligence and Transport*, 14–46. https://doi.org/10.4337/9781803929545.00007

Traffic count. data.wa.gov.au. (n.d.). https://catalogue.data.wa.gov.au/dataset/perth-traffic-count

Traffic volumes aadt. Caltrans Home. (n.d.). https://gisdata-caltrans.opendata.arcgis.com/datasets/d8833219913c44358f2a9a71bda57f76_0/about

Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining kohonen maps with Arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, *4*(5), 307–318. https://doi.org/10.1016/s0968-090x(97)82903-8

Wang, C., Zhang, W., Wu, C., Hu, H., Ding, H., & Zhu, W. (2022). A traffic state recognition model based on feature map and Deep Learning. *Physica A: Statistical Mechanics and Its Applications*, *607*, 128198. https://doi.org/10.1016/j.physa.2022.128198

Wang, Y., Papageorgiou, M., & Messmer, A. (2008). Real-time freeway traffic state estimation based on extended Kalman filter: Adaptive capabilities and real data testing. *Transportation Research Part A: Policy and Practice*, *42*(10), 1340–1358. https://doi.org/10.1016/j.tra.2008.06.001

Wang, Z., Sun, P., Hu, Y., & Boukerche, A. (2022). A novel mixed method of machine learning based models in vehicular traffic flow prediction. *Proceedings of the International Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems on*

International Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems. https://doi.org/10.1145/3551659.3559047

Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transportation Research Record: Journal of the Transportation Research Board*, *1776*(1), 194–200. https://doi.org/10.3141/1776-25

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal Arima process: Theoretical basis and empirical results. Journal of Transportation Engineering, 129(6), 664–672. https://doi.org/10.1061/(asce)0733-947x(2003)129:6(664)

Wu, C.-H., Ho, J.-M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, *5*(4), 276–281. https://doi.org/10.1109/tits.2004.837813

Wu, T., Chen, F., & Wan, Y. (2018). Graph attention LSTM network: A new model for traffic flow forecasting. *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. https://doi.org/10.1109/icisce.2018.00058

Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, *90*, 166–180. https://doi.org/10.1016/j.trc.2018.03.001

Xiao, J., Wei, C., & Liu, Y. (2018). Speed estimation of traffic flow using multiple kernel support vector regression. *Physica A: Statistical Mechanics and Its Applications*, *509*, 989–997. https://doi.org/10.1016/j.physa.2018.06.082

Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *22*(5), 326–334. https://doi.org/10.1111/j.1467-8667.2007.00489.x

Xiong, H., Vahedian, A., Zhou, X., Li, Y., & Luo, J. (2018). Predicting traffic congestion propagation patterns. Proceedings of the 11th ACM SIGSPATIAL International Workshop on Computational Transportation Science. https://doi.org/10.1145/3283207.3283213

Xu, X., Jin, X., Xiao, D., Ma, C., & Wong, S. C. (2021). A hybrid autoregressive fractionally integrated moving average and nonlinear autoregressive neural network model for short-term traffic flow prediction. *Journal of Intelligent Transportation Systems*, *27*(1), 1–18. https://doi.org/10.1080/15472450.2021.1977639

Yang, G., Wang, Y., Yu, H., Ren, Y., & Xie, J. (2018). Short-term traffic state prediction based on the spatiotemporal features of Critical Road sections. *Sensors*, *18*(7), 2287. https://doi.org/10.3390/s18072287

Yang, H.-F., Dillon, T. S., & Chen, Y.-P. P. (2017). Optimized structure of the traffic flow forecasting model with a deep learning approach. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2371–2381. https://doi.org/10.1109/tnnls.2016.2574840

Yang, Z., Lin, C., & Gong, B. (2009). Support Vector Machines for incident detection in urban signalized Arterial Street Networks. 2009 International Conference on Measuring Technology and Mechatronics Automation. https://doi.org/10.1109/icmtma.2009.28

Yao, R., Zhang, W., & Zhang, L. (2020). Hybrid methods for short-term traffic flow prediction based on Arima-GARCH model and Wavelet Neural Network. *Journal of Transportation Engineering, Part A: Systems*, *146*(8). https://doi.org/10.1061/jtepbs.0000388

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A Deep Learning Framework for traffic forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. https://doi.org/10.24963/ijcai.2018/505

Yu, H., Wu, Z., Wang, S., Wang, Y., & Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in Transportation Networks. *Sensors*, *17*(7), 1501. https://doi.org/10.3390/s17071501

Zhang, D., & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: A gru-based deep learning approach. *IET Intelligent Transport Systems*, *12*(7), 578–585. https://doi.org/10.1049/iet-its.2017.0313

Zhang, J., Zheng, Y., Sun, J., & Qi, D. (2020). Flow prediction in spatio-temporal networks based on Multitask Deep Learning. *IEEE Transactions on Knowledge and Data Engineering*, *32*(3), 468–478. https://doi.org/10.1109/tkde.2019.2891537

Zhang, S., & Wang, X. (2013). Human detection and object tracking based on histograms of oriented gradients. 2013 Ninth International Conference on Natural Computation (ICNC). https://doi.org/10.1109/icnc.2013.6818189

Zhang, W., Yu, Y., Qi, Y., Shu, F., & Wang, Y. (2019). Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transportmetrica A: Transport Science*, *15*(2), 1688–1711. https://doi.org/10.1080/23249935.2019.1637966

Zhang, Z., Liu, M., & Xu, W. (2021). Spatial-temporal multi-head attention networks for traffic flow forecasting. Proceedings of the 5th International Conference on Computer Science and Application Engineering. https://doi.org/10.1145/3487075.3487102

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2015). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. JOURNAL OF LATEX CLASS FILES, 14. https://doi.org/arXiv:1811.05320v3

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: A deep learning approach for short-term Traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75. https://doi.org/10.1049/iet-its.2016.0208

Zheng, Z., Yang, Y., Liu, J., Dai, H.-N., & Zhang, Y. (2019). Deep and embedded learning approach for traffic flow prediction in Urban Informatics. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3927–3939. https://doi.org/10.1109/tits.2019.2909904

Zou, Z., Gao, P., & Yao, C. (2018). City-level traffic flow prediction via LSTM Networks. *Proceedings of the 2nd International Conference on Advances in Image Processing*. https://doi.org/10.1145/3239576.3239623