Project Report on

TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEMS USING MACHINE LEARNING

Submitted in the partial fulfillment of the requirements for the Major Project Phase 1 of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted by

B. Deepthi	18071A1206
B. Tarun	18071A1209
J. Siri Smitha	18071A1222
M. Srija	18071A1234

Under the esteemed guidance of

PROJECT GUIDE

Anand Kumar Sharma

Assistant Professor,

Dept. of Information Technology,

VNRVJIET

DEPARTMENT OF INFORMATION TECHNOLOGY

VNR Vignana Jyothi Institute of Engineering & Technology

(Autonomous Institute, Accredited by NAAC with ‘A++’ grade and NBA)

Bachupally, Nizampet (S.O.) Hyderabad- 500 090,

2021- 2022

VNR Vignana Jyothi Institute of Engineering & Technology

Autonomous Institute, Accredited by NAAC with ‘A++’ grade and
NBA) Bachupally, Nizampet (S.O.) Hyderabad- 500 090

Department of Information Technology

CERTIFICATE

This is to certify that the project work entitled **“TRAFFIC PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM USING MACHINE LEARNING”** is being submitted by **B. Deepthi (18071A1206), B. Tarun (10871A1209), J. Siri Smitha (18071A1222), M. Srija (18071A1234)** in partial fulfillment for the award of Degree of **BACHELOR OF TECHNOLOGY** in **INFORMATION TECHNOLOGY** to the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2021-2022 is a record of bonafide work carried out by her under our guidance and supervision.

The results embodied in this report have not been submitted by the students to any other University or Institution for the award of any degree or diploma.

Project Guide	**Head of Department**
ANAND KUMAR SHARMA,	**Dr. D. SRINIVASA RAO**
Assistant Professor,	**Head of Department,**
Dept. of IT,	**Dept. of IT,**
VNRVJIET,	**VNRVJIET,**
Hyderabad.	**Hyderabad.**

VNR Vignana Jyothi Institute of Engineering & Technology
Autonomous Institute, Accredited by NAAC with ‘A++’ grade and
NBA) Bachupally, Nizampet (S.O.) Hyderabad- 500090.

Department of Information Technology

DECLARATION

We hereby declare that the project entitled “Traffic Prediction for Intelligent Transportation System using Machine Learning” submitted for the B.Tech Degree is our original work and the project has not formed the basis for the award of any degree, associateship, fellowship, or any other similar titles.

Signature of the Student:

B. Deepthi	B. Tarun	J. Siri Smitha	M. Srija
(18071A1206)	(18071A1209)	(18071A1222)	(18071A1234)

Place:
Date:

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved **President, Mr. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering &Technology** for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved **Principal, Dr. C. D. Naidu** for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved professor **Dr. D. Srinivasa Rao, Associate Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-90** for the valuable guidance and suggestions, keen interest, and encouragement extended throughout project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide **Anand Kumar Sharma, Assistant Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad,** for his valuable suggestions and rare insights, for the constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed to the successful completion of our project work till the design phase (Phase 1).

1. B. Deepthi (1801A1206)
2. B. Tarun (18071A1209)
3. J. Siri Smitha (18071A1222)
4. M. Srija (18071A1234)

TABLE OF CONTENTS

ABSTRACT

Traffic flow prediction is one of the key problems in traffic control and guidance system as well as the important functions of intelligent transportation systems (ITS). Traffic estimation and prediction systems can reduce traffic congestion and improve road capacity effectively. Due to the fast economic growth and the highly increasing number of vehicles, the first challenge is to successfully predict accurate traffic flow information to minimize traffic congestion and traffic accidents. Not long ago, many researchers have started to focus more and more concentration on deep learning techniques, including Recurrent Neural Networks (RNN), especially due to their capacity to learn long-term dependencies of sequence data and capture the nonlinearity nature of traffic flow. In this project, we aim to improve the accuracy of traffic flow prediction by applying three different kinds of recurrent neural network architecture such as simple RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), and compare the performance of all 3 algorithms. We would be using the dataset collected from the California department of transportation combining it with vehicle speed and weather conditions dataset to improve the accuracy obtained. To evaluate the efficiency of the machine learning model we use two popular metrics, including Mean Absolute Percentage Errors (MAPE) and Root Mean Squared Error (RMSE).

1: INTRODUCTION

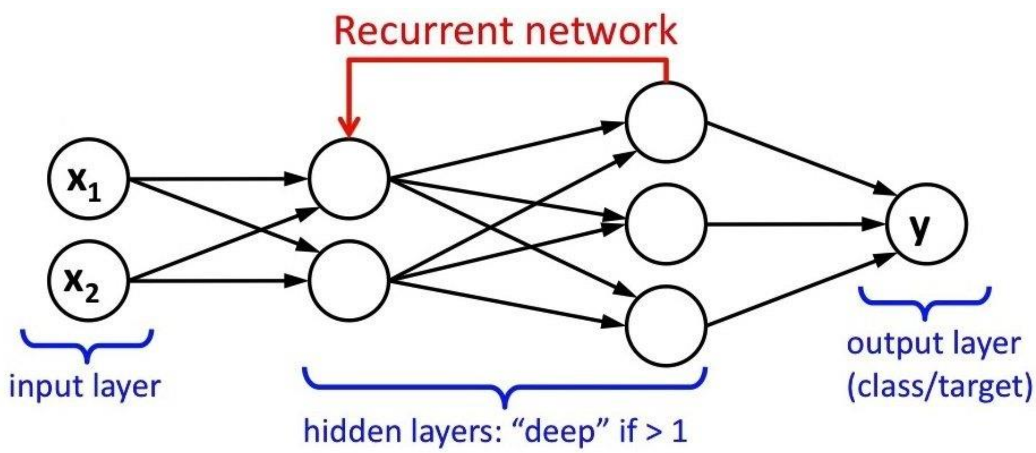
The modern city is gradually developing into a smart city. The acceleration of urbanization and the rapid growth of the urban population bring great pressure to urban traffic management. Intelligent Transportation System (ITS) is an indispensable part of smart city, and traffic prediction is an important component of ITS. Accurate traffic prediction is essential to many real-world applications. For example, traffic flow prediction can help the city alleviate congestion; car-hailing demand prediction can prompt car-sharing companies pre- allocate cars to high-demand regions. The growing available traffic-related datasets provide us with potential new perspectives to explore this problem.

With the progress of urbanization and the popularity of automobiles, transportation problems are becoming more and more challenging: traffic flow is congested, accidents are frequent, and the traffic environment is deteriorating. The question of how to improve the capacity of the road network has attracted attention from an increasing number of scholars. But traffic congestion is not primarily a problem but a solution to our basic mobility problem which is that too many people want to move at the same time each day. Because the efficient operation of both economy and school systems requires that people work, go to school, and even run errands during about the same hours so that they can interact with each other. An intelligent transportation system is used for analyzing information. ITS is used to control communication technologies for road transportation to improve safety and efficiency. Intelligent transportation system includes a wide range of applications which is used to get information, control congestion, improve traffic management, reduce the environmental effects, and increase the benefits of transportation. ITS refers to the different types of needs and the transport field with many others policing. But also due to less connection of traffic flow. Smartphones have different sensors. It can be used to detect/track traffic speed and density. Nowadays, smartphones are used by drivers, and it is monitored to detect the speed of traffic and quality of the road. Data is connected through audio and GPS.it tracks the identity of traffic and possible jams that occurred in the traffic.

Although precise traffic prediction is a huge problem to solve, the massive traffic data collected holds missing values or incorrect values for many reasons like equipment errors and incorrect measurement, leading to an inaccurate prediction and poor-quality output. One of the best solutions to such imperfections is data preprocessing in which the dataset is prepared and cleaned.

RNN (Recurrent Neural Networks):

One of the most difficult types of data to handle and the forecast is sequential data. Sequential data is different from other types of data in the sense that while all the features of a typical dataset can be assumed to be order-independent, this cannot be assumed for a sequential dataset. To handle sequential data, the concept of Recurrent Neural Networks was conceived. It is different from other Artificial Neural Networks in its structure. While other networks “travel” in a linear direction during the feed-forward process or the back-propagation process, the Recurrent Network follows a recurrence relation instead of a feed-forward pass and uses Backpropagation through time to learn. A Recurrent Neural Network (RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs. The main and most important feature of RNN is the Hidden state, which remembers some information about a sequence. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, or speech recognition.



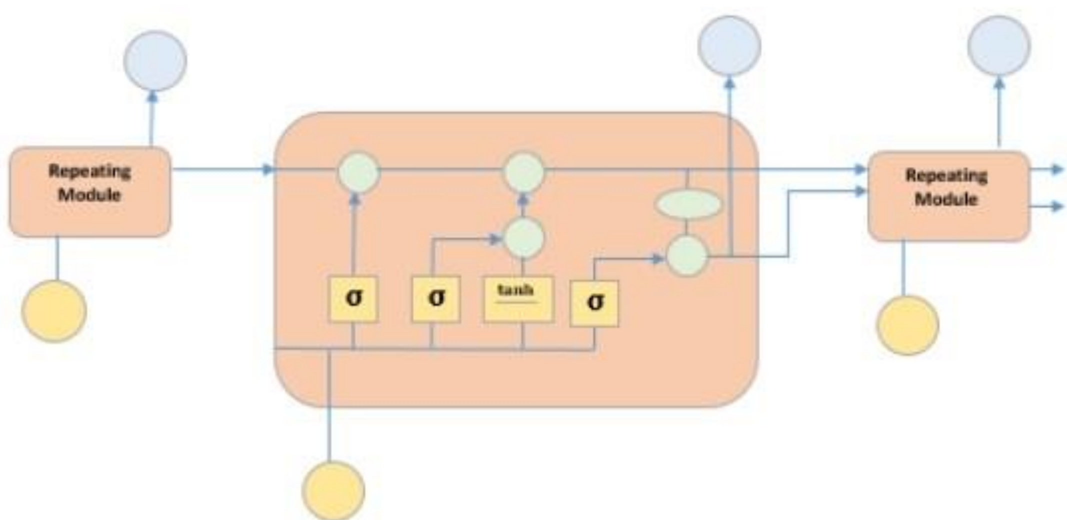
The Recurrent Neural Network consists of multiple fixed activation function units, one for each time step. Each unit has an internal state which is called the hidden state of the unit. This hidden state signifies the past knowledge that the network currently holds at a given time step. This hidden state is updated at every time step to signify the change in the knowledge of the network about the past.

RNN has a "memory" which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks. In an RNN the information cycles through a loop. When it decides, it considers the current input and what it has learned from the inputs it received previously. Therefore, an RNN has two inputs: the present and the recent past.

LSTM (LONG SHORT-TERM MEMORY):

Long short-term memory networks (LSTMs) are an extension for recurrent neural networks, which extend the memory. In RNN output from the last step is fed as input in the current step. LSTM tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period. LSTMs assign data "weights" which helps RNNs to either let new information in, forget information, or give it important enough to impact the output. It is used for processing, predicting, and classifying based on time-series data. Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs.

The units of an LSTM are used as building units for the layers of an RNN, often called an LSTM network. LSTMs enable RNNs to remember inputs over a long period. This is because LSTMs contain information in a memory. The LSTM can read, write, and delete information from its memory. This memory can be seen as a gated cell, with gated meaning the cell decides whether or not to store or delete information based on the importance it assigns to the information. The assigning of importance happens through weights, which are also learned by the algorithm. This simply means that it learns over time what information is important and what is not.



Structure Of LSTM:

LSTM has a chain structure that contains four neural networks and different memory blocks called cells. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. Information is retained by the cells and the memory manipulations are done by the gates. In an LSTM you have three gates: input, forget and output gate. These gates determine whether to let new input in (input gate), delete the information because it isn't important (forget gate), or let it impact the output at the current timestep (output gate). The gates in an LSTM are analog in the form of a sigmoid, meaning they range from zero to one. The fact that they are analog enables them to do backpropagation. The problematic issues of vanishing gradients are solved through LSTM because it keeps the gradients steep enough, which keeps the training relatively short and the accuracy high. The information at a particular cell state has three different dependencies.

1. The previous cell state (i.e. the information that was present in the memory after the previous time step)

2. The previous hidden state (i.e. this is the same as the output of the previous cell)

3. The input at the current time step (i.e. the new information that is being fed in at that moment)

LSTM works in four steps:

1. Information to be forgotten is identified from the previous time step using forget gate.

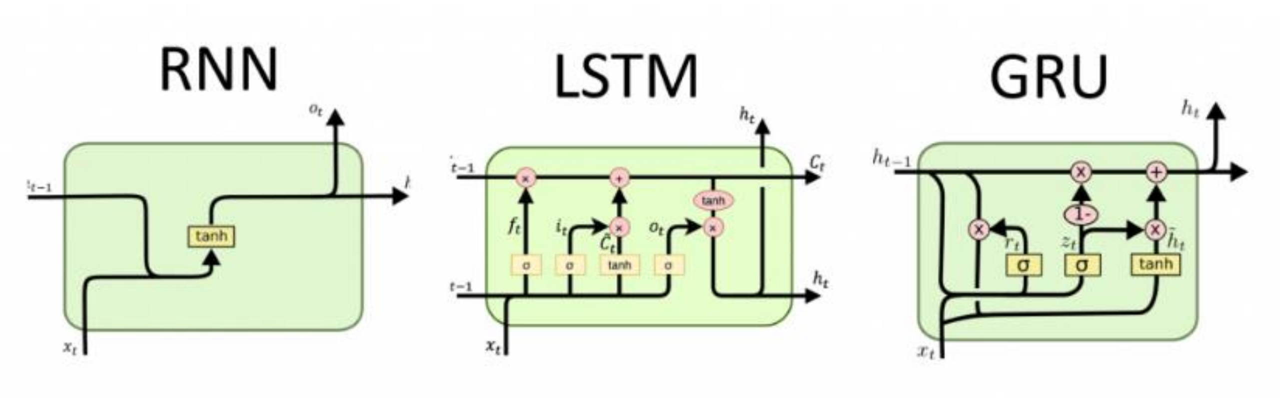
2. New information is sought for updating cell state using input gate and tanh.

3. Cell state is updated using the above two gates information.

4. Relevant information is yielded using the output gate and the squashing function

GRU (Gated Recurrent Unit):

GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results. To solve the vanishing gradient problem of a standard RNN, GRU uses an update gate and reset gate. These are two vectors that decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or removing information that is irrelevant to the prediction. Using these two vectors, the model refines outputs by controlling the flow of information through the model. Like other kinds of recurrent network models, models with gated recurrent units can retain information over a period – that is why one of the simplest ways to describe these types of technologies is that they are a "memory-centered" type of neural network. By contrast, other types of neural networks without gated recurrent units often cannot retain information. In addition to speech recognition, neural network models using gated recurrent units may be used for research on the human genome, handwriting analysis, and much more. Some of these innovative networks are used in stock market analysis and government work. Many of them leverage the simulated ability of machines to remember information. The two gates including a reset gate that adjusts the incorporation of new input with the previous memory and an update gate that controls the preservation of the precious memory are introduced. The reset gate and the update gate adaptively control how much each hidden unit remembers or forgets while reading/generating a sequence.



Through this project, we would evaluate different models like the LSTM model, GRU, and Simple RNN and compare the performance of all 3 models and establish the best algorithm for predicting traffic data. And finally, With the help of the most accurate model, we will build a web page where a user can input two places (the source and the destination), and the model would return how long it would take for them to reach their destination with the traffic conditions (mild, high, low). The major advantage of this model is that we can forecast the traffic conditions for 8 to 12 hours.

1.1 DEFINITION:

The traffic flow congestion and traffic data in modern areas have been blown up in the past years because of the rising number of cars. People get hit in the traffic for many hours, so individual travelers and Intelligent Transportation System (ITS) precise that traffic flow is important for both drivers. With different advanced technologies used these days, electronic devices are being deployed to collect traffic data such as passing vehicle details, including volumes, speed, and class at a certain time. However, it is possible to use the detailed reviewed data collected to help transport planners improve existing road networks or construct new ones based on the predicted long-term and short-term traffic flow. All of these are in ITS, the traffic prediction foundational.

1.2 OBJECTIVE:

The main objective of our project is to predict traffic. To do so we will check for the algorithm which gives the accurate Traffic Prediction results by applying different Machine Learning algorithms like GRU, RNN, and LSTM and comparing their performance to establish the best model. Using the dataset collected from the California department of transportation and combining it with vehicle speed and weather conditions dataset we aim to improve the accuracy obtained. With the help of the most accurate algorithm, we would like to make it user friendly by creating a webpage so that the users can input two places (the source and the destination), and the model would return how long it would take for them to reach their destination with the traffic conditions (mild, high, low). The major advantage of this model is that we can forecast the traffic conditions for 8 to 12 hours.

2: LITERATURE SURVEY

After years of effort, the research on traffic prediction has achieved great progress. Considering the development process, these methods can be broadly divided into two categories: classical methods and deep learning-based methods. Classical methods include statistical methods and traditional machine learning methods. The statistical method is to build a data-driven statistical model for prediction. The most representative algorithms are Historical Average (HA), Auto-Regressive Integrated Moving Average (ARIMA) [16], and Vector Auto-Regressive (VAR) [17]. Nevertheless, these methods require data to satisfy certain assumptions, and time-varying traffic data is too complex to satisfy these assumptions. Moreover, these methods are only applicable to relatively small datasets. Later, several traditional machine learning methods, such as Support Vector Regression (SVR) [18] and Random Forest Regression (RFR) [19], were proposed for traffic prediction problems. Such methods can process high-dimensional data and capture complex non-linear relationships.

Techniques used for traffic forecasting have steadily shifted from statistical models to machine learning intelligence and have been into two major classes which involved parametric and non-parametric models [5–7]. Furthermore, due to stochastic and nonlinear traffic flow characteristics, the parametric linearity method did not provide high efficiency in predicting the next situations and more Researchers started to concentrate on the non-parametric methods which try to learn historical data which is related to the expectation instant and use the information items found to forecast for the future.

Researchers have presented many traffic flow forecasting approaches whereby they made attention to short-term traffic flow prediction but is still observed as a challenge today [8]. According to the literature in parametric models, Autoregressive Integrated Moving Average (ARIMA) and its variants are one of the most consolidated approaches based on classical statistics and have been widely applied for traffic prediction problems ([1], [2],[3]) One weakness of ARIMA is its inherent propensity to focus on the data's mean values from the past sequence. It remains difficult, therefore, to capture a rapidly changing phase [9]. Because of the failure due to nonlinear and stochastic parametric models which are not able to predict accurately, non-parametric models have been studied and built by more researchers including the Support Vector Regression

(SVR)application successfully submitted for the prediction of time series and has shown some disadvantages, such as the lack of standardized means to decide some primary model parameters [9]. Neural networks implementations have become the latest interest in the traffic research field.

The contrast between traditional models and neural networks distinctly presents an upper level in predicting accurate traffic information [10]. one of the deep learning models called Recurrent Neural Network (RNN) establish the reputation for dealing with time series via recurrent neural ties; however, Gers et al. in [11,12] show that firstly there are still many problems to be tackled in fashion because RNNs do not train with long time lags in the time series, although this incident is commonly seen in traffic prediction tasks. Secondly, to learn the processing of the temporal series, RNNs rely on predetermined time lags, but it is not easy to Find in an automated way the optimum time window size. By altering the arrangement of the secret neurons in conventional RNN, Long Short-Term Memory (LSTM) has been revolutionarily designed to solve the problem. Wang et al. [13] apply LSTM based approach for the next moment prediction of traffic load in a particular geometric field. In [14], LSTM was applied for traffic speed prediction with remote microwave sensor data. Yongxue Tian and Li pan [8] compared different models including SVM, SAE, FFNN, and LSTM RNN, and conclude that the LSTM RNN model achieves the best results between these non-parameter models. Li et al. assessed the LSTM and GRU model efficiency to predict traffic flow [15].

The review process is divided into five stages to find the process in a simple and adaptable way. It is necessary to start with a particular domain of any division/city of interest and it causes a specific problem. Literature also tells us that AVs would reduce vehicle ownership, travel timing, parking lots, and emissions It is also telling that AVs would increase the road capacity, traffic flow stability, vehicle miles traveled, fuel efficiency, and safety. The ACC (Adaptive Cruise Control) can perform total control of the vehicle by focussing on the speed without any data from the driver/conductor.

The main aim of the Smart transport system is to distribute new services to various modes of transportation and manage traffic. The flow of traffic forecasting is the key attribute used to manage traffic. Due to the growth of modern technology in real-time, various new techniques and types of equipment are used in the traffic prediction system. From the various new techniques, deep learning is one of the important concepts used to retrieve important characteristics effectively from the amount of raw data with the help of unseen layers. Traffic data with nonlinear features is one of the important reasons for producing a less accurate result in traffic prediction. Shiju George et al., 2020 propose a new bioinspired technique with a fuzzy concept. Technological indicators issue the flow characteristics of an input. Unseen layers of the Deep learning framework continuously learn the characteristics and transmit them to the next level layer. Membership degree is measured with the help of membership methods. Finding optimized weight value with the help of the Dolphin Echolocation concept to set the model for data with nonlinear features. Experiments were conducted on two various datasets and displayed the output for the new proposed deep learning-based framework. Produced results show the key significance of traffic jam prediction.

Vehicle identification is an important technique in the transport system. By identifying the vehicle, the number of automobiles is known and the presences of the vehicles on the path are important factors. High-dimensional data can be used to denote automobiles. Feature retrieval and classifying features are the important processes used to identify automobiles. High dimension data takes more computation time during the feature extraction process. D. M. S. Arsa et al., 2017, propose the DBN (Deep Belief Network) technique for reducing the dimension of the data to detect the vehicles. In this research, the authors try to identify motorbikes and cars. Here DBN technique is used to reduce the dimension of the data and the SVM concept is used to classify the data. The proposed method is applied to the UIUC dataset and the outcome of the current technique is compared with the PCA method. The experiment outcomes show that the DBN concept provides a better result than the traditional PCA method in the identification of automobiles.

In the transport system traffic flow forecasting is a major issue. Various existing techniques produce unsuccessful output due to various reasons like thin framework, engineering manually, and learning separately. W. Huang et al., 2014 propose a new deep framework that contains two basic parts. The base part contains DBN and the top portion contains the regression layer. DBN technique is applied for the purposed of unsupervised learning and it learns efficient attributes for traffic forecasting in an unsupervised manner. It produces a better result for many places like audio and image classification. The regression layer is used for supervised forecasting. The experiment outcomes describe that the proposed method increases the performance of existing systems. The positive outcomes say that multitask regression and deep learning are important technologies in transport system research.

Sheik Mohammed Ali et al., 2012 presented the best algorithm to categorize automobiles recognized using many inductive systems. RF (Random Forest) algorithm can be used for classification purposes. This proposed system is used to categorize the vehicles and count them based on the traffic situation. The output of the proposed method compared with other techniques depends on signature and threshold data. The outcome from the proposed system shows improved accuracy compared to signature and threshold-based techniques.

According to Zhenbo Lu et al., 2019 recognizing and identifying the mode of travel and passenger travel pattern are the major issues in the transportation system. MSD (Mobile- phone Signaling Data) technique has various merits like wide area coverage and less acquisition amount, reliability and stability of data, and better performance in real-time. Here the authors develop a travel mode identification system using MSD integrated with travel data. GIS data and navigation type data. The proposed system is applied to the Kunshan data set in China and the model produces better accuracy of 90%. This accuracy level is suitable for all kinds of transport modes except for buses.

3: EXISTING SYSTEM

There have been many types of research conducted in the area - traffic prediction. Most of the existing research employs various machine learning algorithms and other deep learning techniques such as RNN, CNN, and ARIMA models.

But most of the existing systems are only at the level of a single data source, they don't consider other factors such as accidents, weather data, and vehicle speed.

A lot of the datasets present online have missing values due to incorrect measurement and equipment errors.

3.1 EXISTING SYSTEM DRAWBACKS:

- Most existing solutions are data intensive. However, abnormal conditions (extreme weather, temporary traffic control, etc.) are usually non-recurrent.
- At present, traffic prediction still mainly stays at the level of a single data source, with less consideration of influencing factors. With more collected datasets, we can obtain more influencing factors.
- In the process of collecting traffic data, due to factors such as equipment failures, the collected information deviates from the true value. The use of contaminated data for modeling will affect the prediction accuracy of the model
- At the same time, most existing technologies are based on short-term interval prediction, we aim to predict at least for 8 to 12 hours.

4: PROPOSED SYSTEM

Any technique that uses Machine Learning technologies can be divided into small functions:

A. Collection of Data

B. Processing of Data

C. Decision Making System

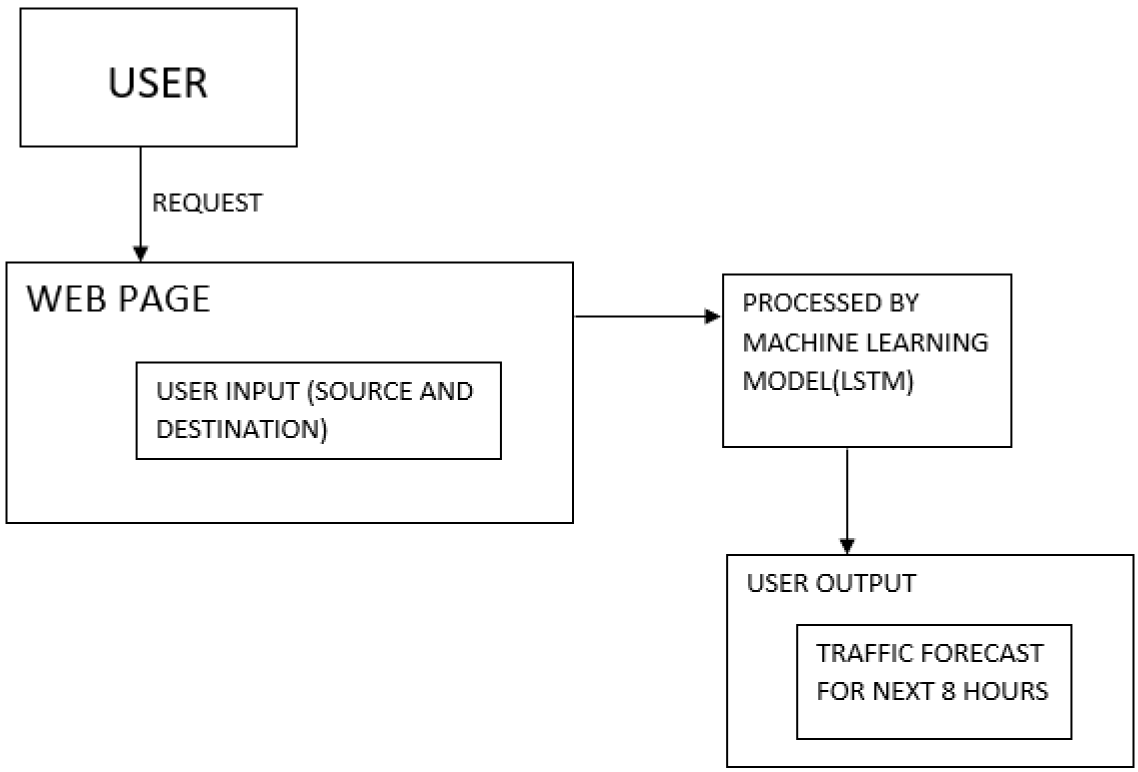
In this project, we aim to improve the accuracy of traffic flow prediction by applying three different kinds of recurrent neural network architecture such as simple RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), and compare the performance of all 3 algorithms. We would be using the dataset collected from the California department of transportation combining it with vehicle speed and weather conditions dataset to improve the accuracy obtained.

To evaluate the efficiency of the machine learning model we use two popular metrics, including Mean Absolute Percentage Errors (MAPE) and Root Mean Squared Error (RMSE).

With the most accurate algorithm, we would like to make it user friendly by creating a webpage so that the users can input two places (the source and the destination), and the model would return how long it would take for them to reach their destination with the traffic conditions (mild, high, low). The major advantage of this model is that we can forecast the traffic conditions for 8 to 12 hours.

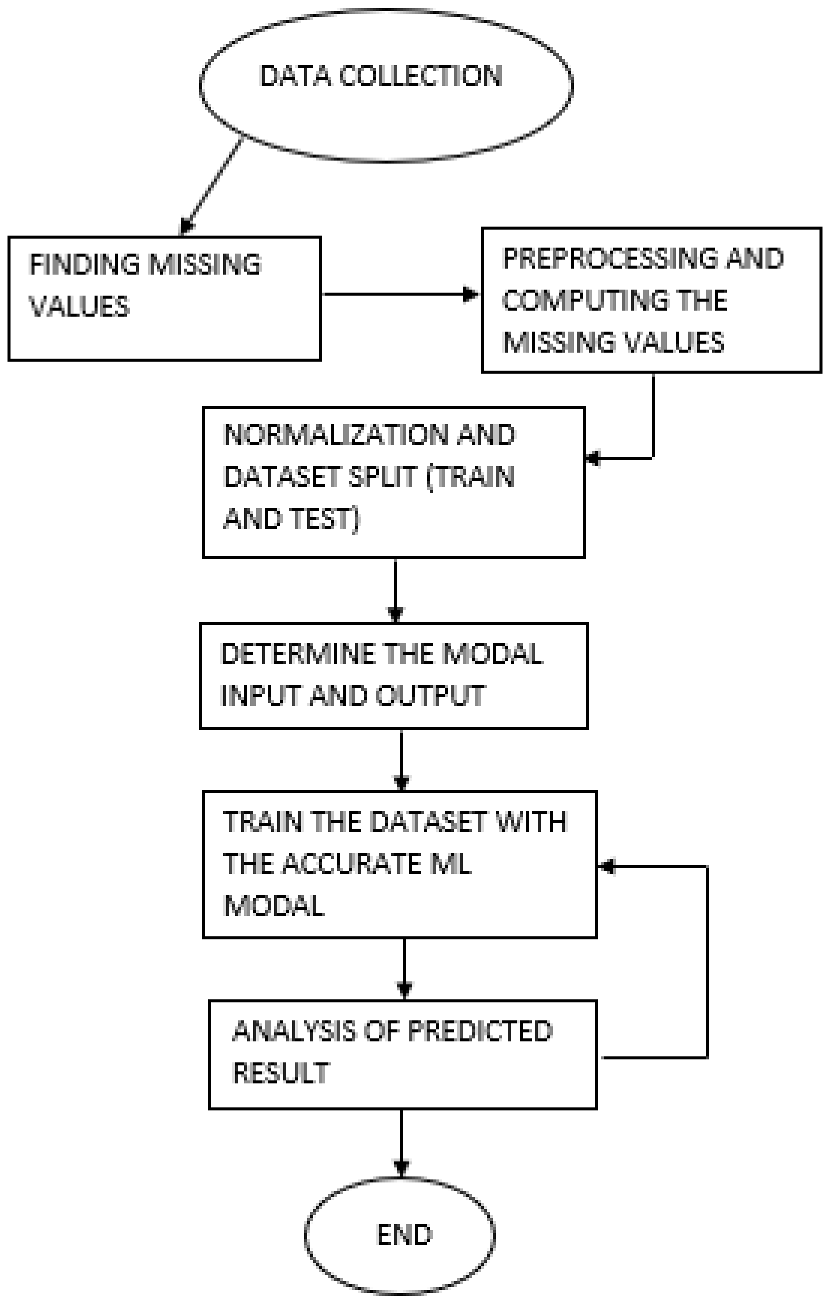
5: ARCHITECTURE

Fig 1: System Architecture Diagram



The system architecture diagram depicts the overall outline of the software system and the relationships, constraints, and boundaries between components. When the user opens the website, the main screen will be displayed which contains place it types in 2 inputs – source, destination. After the user types in his source and destination, the machine learning model would forecast traffic for the next 8 hours and give you the traffic density per hour.

FLOW CHART FOR MACHINE LEARNING MODAL:



6: MODULES

The traffic prediction system is an application that focuses on forecasting the traffic conditions for the next 6 to 8 hours. It is a prototype of a new product that comprises two main modules:

Machine-learning module and user interface module

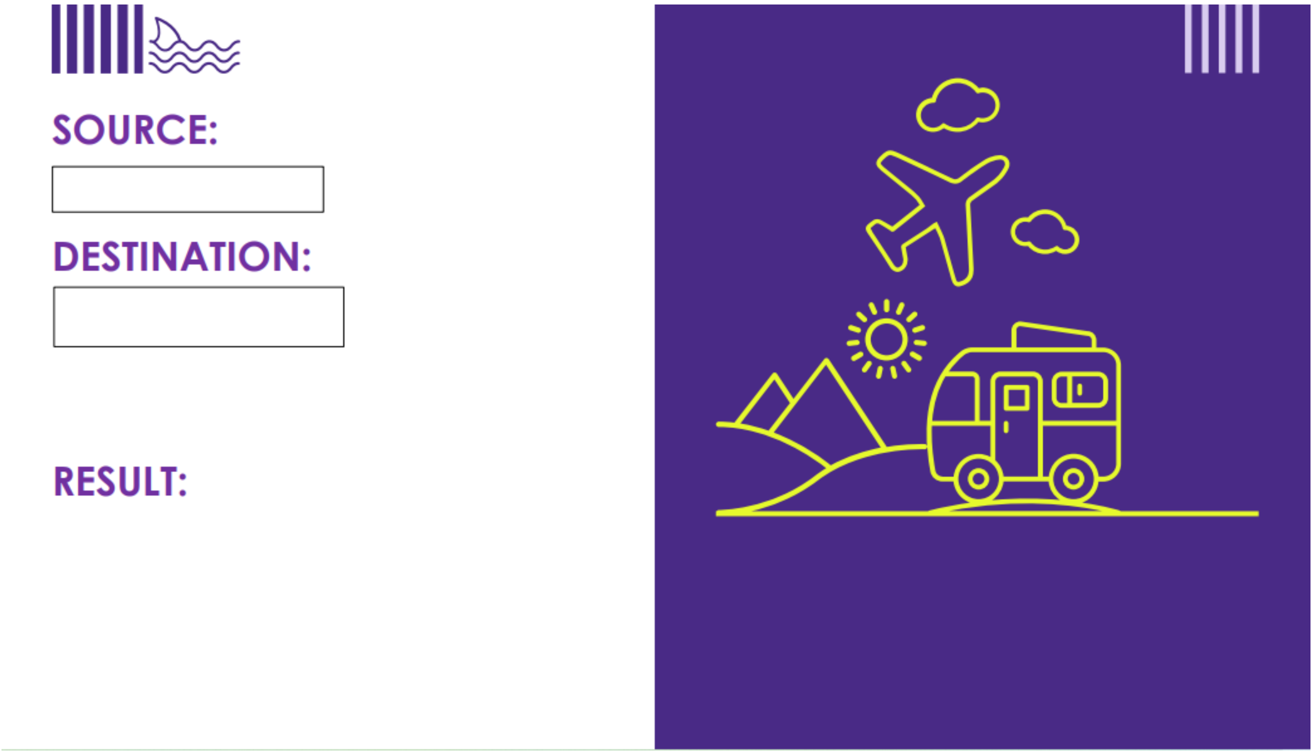
A. Machine learning Module

In this module, various datasets are collected from reliable sources. Datasets such as the traffic dataset, vehicle speed, and weather dataset are gathered. For our predictions to be closer to reality, all these datasets need to aggregate. After which we can go about our regular process. This is the meat of the project after datasets are collected, the required pre-processing is done to make it applicable to apply various machine learning algorithms. Various python libraries are used in this step (Pandas, Numpy, Matplotlib, seaborne). Next, the model is split into training, testing, and validation data in the ratio 70:20:10. Then the model is trained for various algorithms like RNN, GRU, and LSTM. Tensorflow and keras are used. We calculate the RMSE scores and the R2 scores of each algorithm and compare them and determine which algorithm is the best for predicting highly unpredictable data like traffic. A lower RSME score indicates a better-fitted model and the higher the R2 score, indicates that the better the model fits your data. After we have determined which algorithm works best, then we use that algorithm to forecast the traffic conditions for the next 8 hours and display it in the form of a graph.

B. User interface Module

To make it accessible to users, we create a website, where the user can enter two inputs (the source and the destination) and the model would output the traffic forecast for the next 8 hours. This way the data would be more presentable, and users can easily check the traffic predictions before heading out on a journey and get a random estimate of how long it will take them to reach their destination and the traffic density per hour.

To create this website, we make use of front-end and back-end technologies like HTML, CSS, JavaScript, NodeJS, Mysql, e.t.c



7: FEASIBILITY STUDY

Even though traffic data is complex and highly unpredictable, a machine learning model can be trained to accurately predict traffic density from one place to another.

The various data collected and aggregated together would provide us with a more reliable dataset, which we can use to predict traffic conditions. We would assign 3 numbers attributed to 3 traffic density options (high, mild, low). When traffic is forecast then the model would return one of the 3 options. For all this to become user-friendly we create a website wherein the user can type in 2 inputs (the source and the destination) and the model would output the forecast for the next 8 hours and tell you about the traffic density per hour.

A lot of the data collected contains missing values or errors due to equipment failures or other such reasons. The challenge here is to fill in those missing values with accurate values so that the accuracy is not compromised. But this can be more difficult than seems because traffic is highly unpredictable and to be able to use regular methods like filling in the mean or the mode will not be applicable.

Even after we have figured out a way to fill in the missing values, we still cannot be sure of the results, because traffic as it is, is highly unpredictable. Thus, for more accurate results it needs to be trained with more data which includes accidents, vehicle speed, weather data e.t.c for more accurate predictions. To be able to predict /forecast traffic one day before traveling to a particular destination is a possible future scope of our project. Our project doesn't predict real-time data and that could also be considered as the future scope of this project.

8: SOFTWARE AND HARDWARE REQUIREMENT

8.1 Software Requirements:

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements that are required for this project are:

- Google Colab/ Jupiter Notebook
- Machine Learning
- Python
- Front end and Back end (for webpage)

8.2 Hardware Requirements:

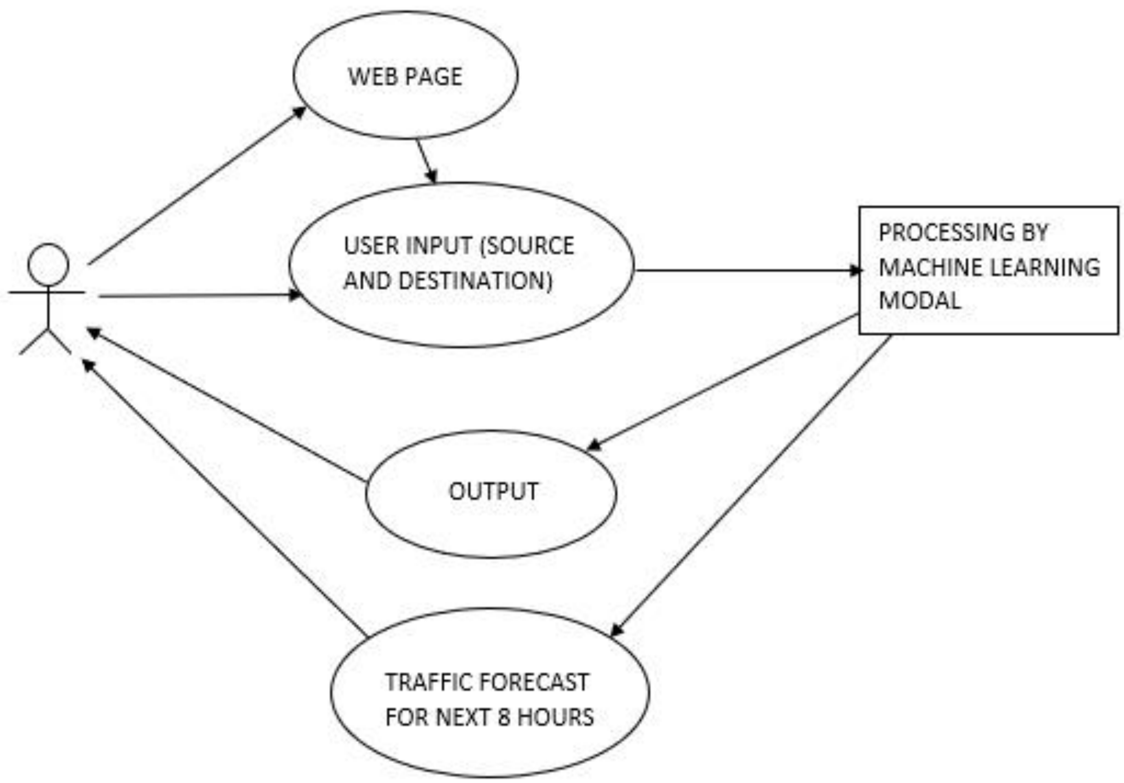
The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The hardware requirements required for this project are:

- Laptop windows 10

9: DESIGN

USE CASE DIAGRAM:

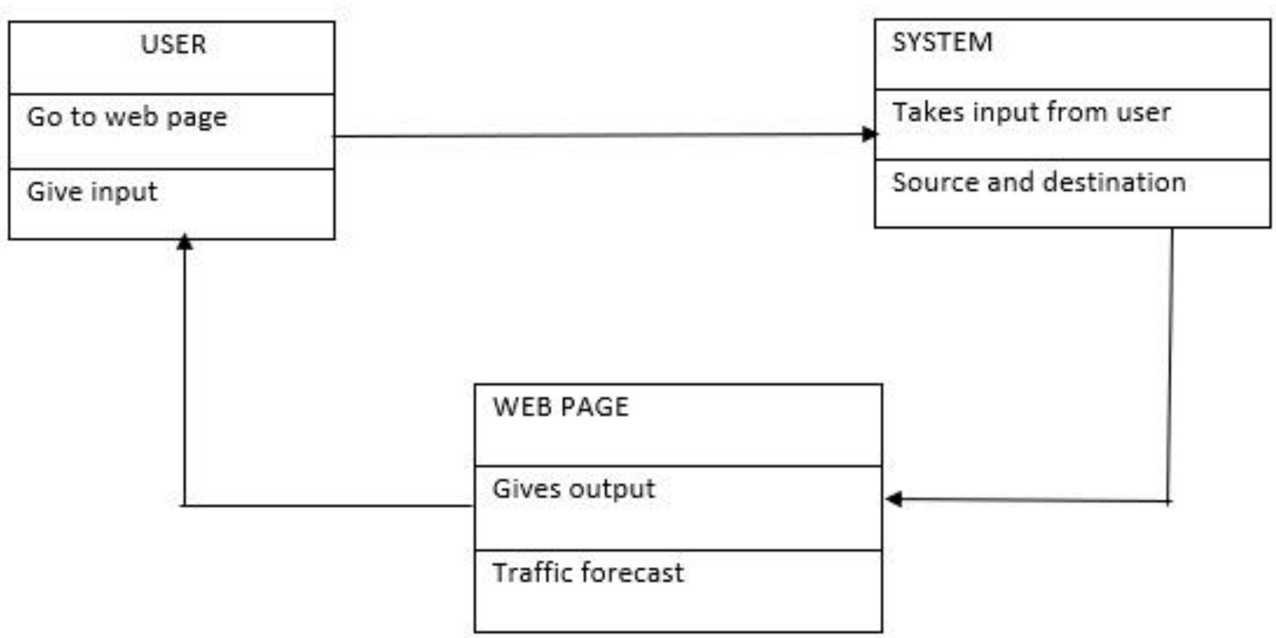
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and tells how the user handles a system.



CLASS DIAGRAM:

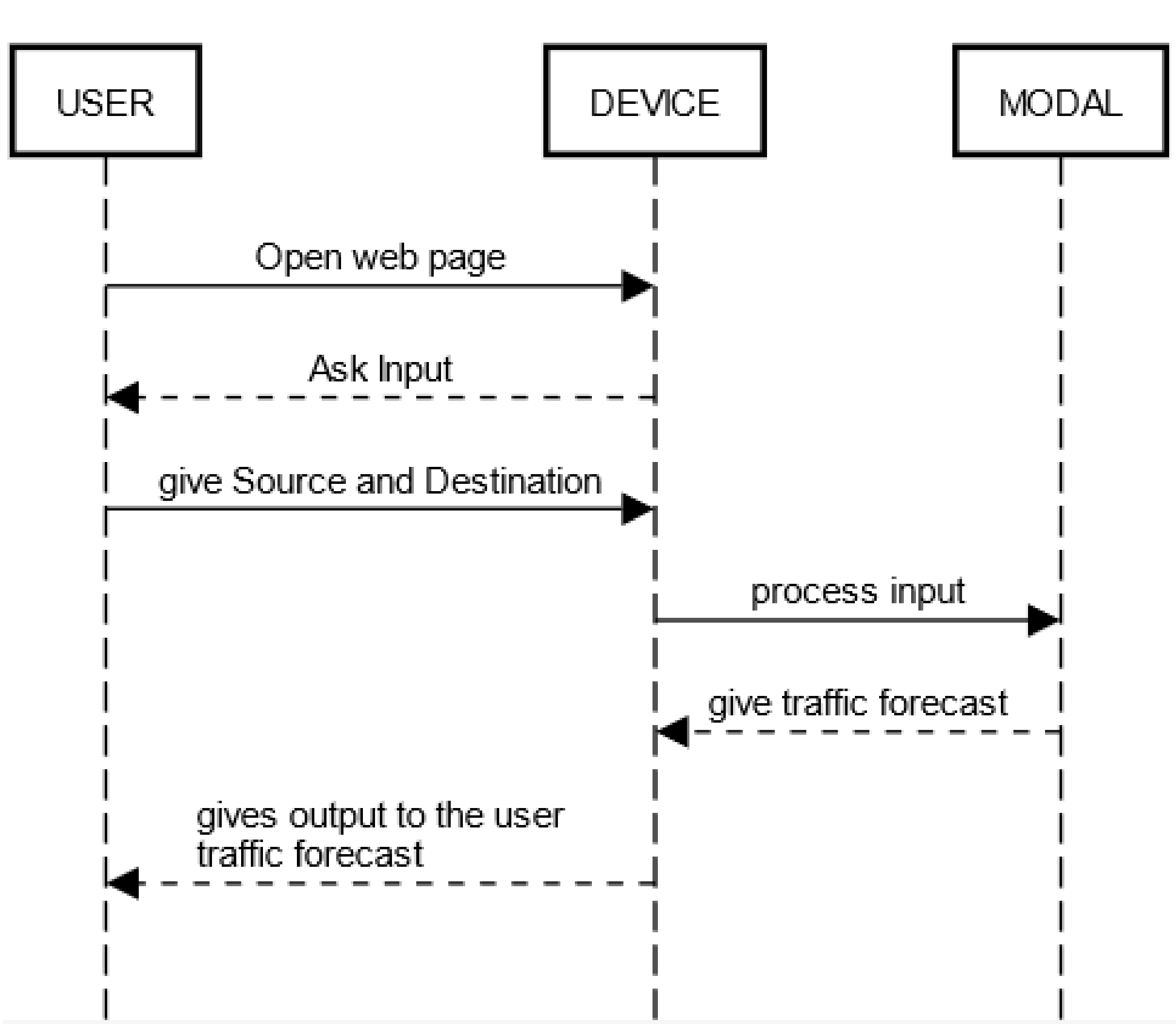
A class diagram is a static diagram. It represents the static view of an application. The class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

A class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.



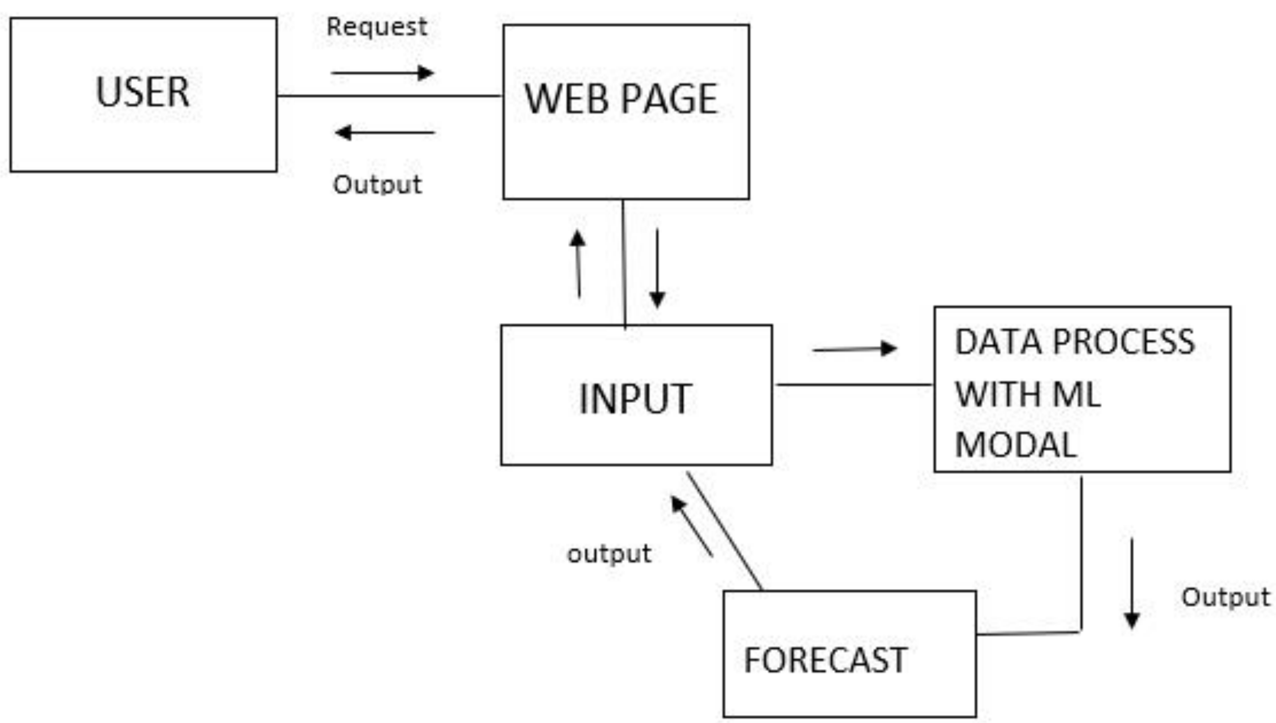
SEQUENTIAL DIAGRAM:

A sequence diagram simply depicts the interaction between objects in sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.



COLLABORATIVE DIAGRAM:

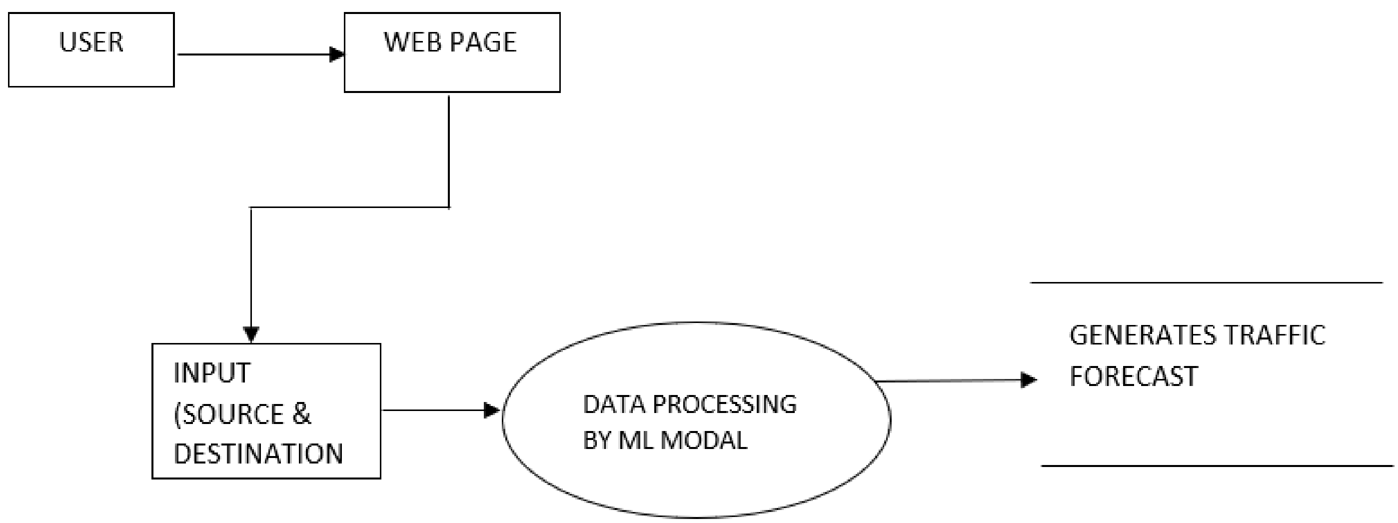
The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.



STATECHART DIAGRAM:

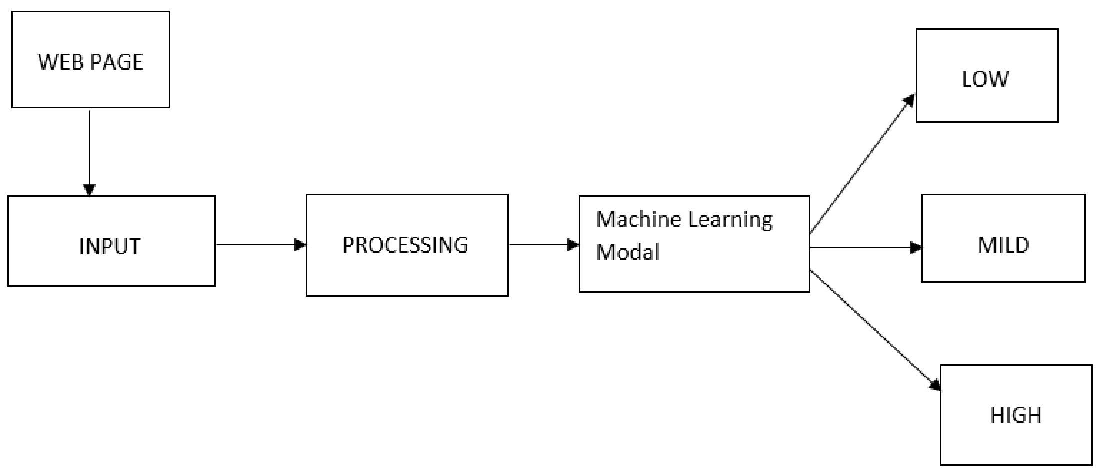
Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists, and it changes when some event is triggered. The most important purpose of the Statechart diagram is to model the lifetime of an object from creation to termination.



ACTIVITY DIAGRAM:

We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.



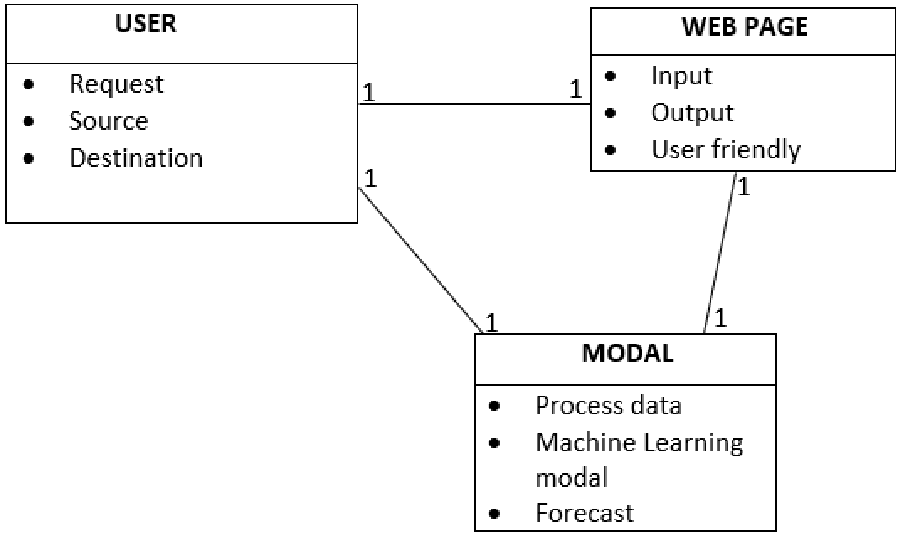
COMPONENT DIAGRAM:

The component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system, but it describes the components used to make those functionalities.

Thus, from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

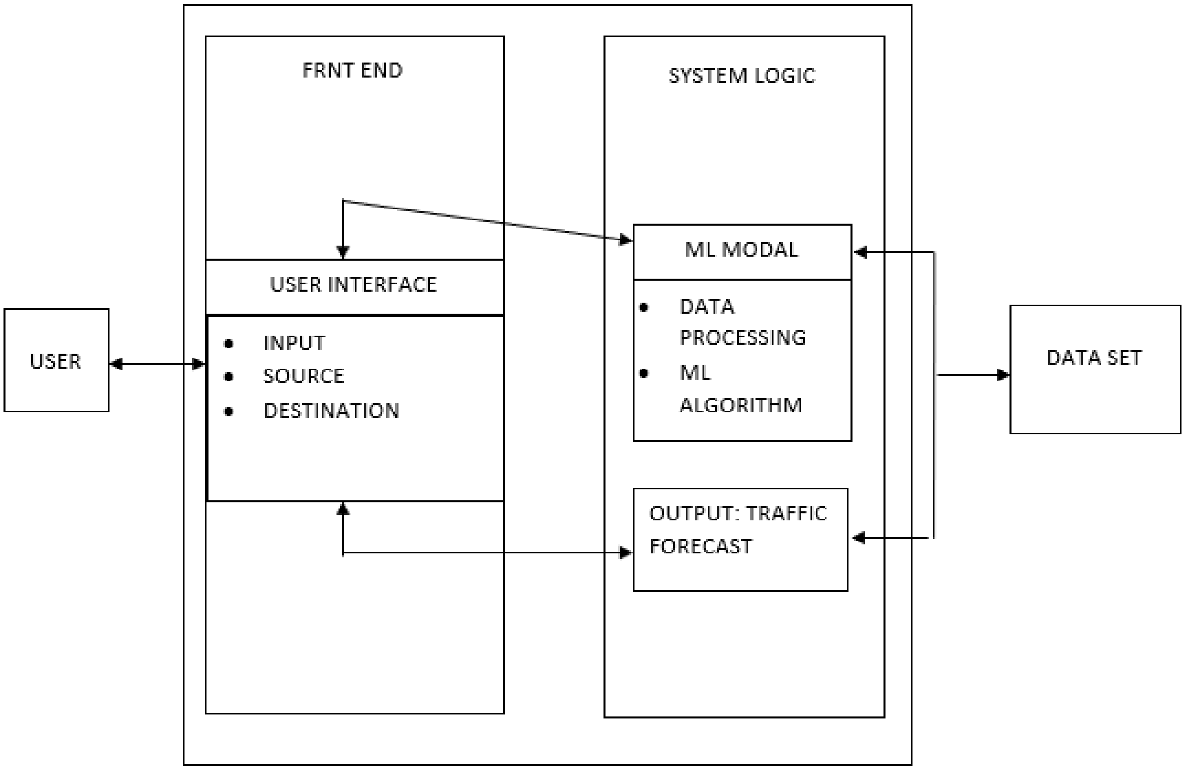
A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.



DEPLOYMENT DIAGRAM:

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



10: CONCLUSION

Through this project, we aim to establish the best algorithm to predict complex data like traffic, by making comparisons between various machine learning algorithms. And finally, we also build a webpage to make it user-friendly using front-end back-end technologies like HTML, CSS, JavaScrip, Mysql, etc, where users can give two inputs - the source and the destination and the model would return the output the traffic forecast for the next 8 hours and give with the traffic conditions (mild, high, low) per hour.

11: REFERENCE

[1] B. Williams and L. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” Journal of transportation engineering, vol. 129, no. 6, pp. 664– 672, 2003.

[2] S. Shekhar and B. Williams, “Adaptive seasonal time series models for forecasting short-term traffic flow,” Transportation Research Record, vol. 2024, no. 1, pp. 116–125, 2007.

[3] I. Wagner-Muns, I. Guardiola, V. Samaranayke, and W. Kayani, “A functional data analysis approach to traffic volume forecasting,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 878–888, 2017.

[5] A. Singh, A. Shadan, R. Singh, and Ranjeet, “Traffic forecasting,” International Journal of Scientific Research and Review, vol. 7, no. 3, pp. 1565–1568, 2019.

[6] A. Boukerche and J. Wang, “Machine learning-based traffic prediction models for intelligent transportation systems,” Computer Networks, vol. 181, p. 107530, 2020.

[7] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, “Road traffic forecasting: Recent advances and new challenges,” IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 2, pp. 93–109, 2018.

[8] Y. Tian and L. Pan, "Predicting Short-term Traffic Flow by Long Short-Term Memory Recurrent Neural Network," 2015, doi: 10.1109/SmartCity.2015.63.

[9] W. Hong, "Application of seasonal SVR with a chaotic immune algorithm in traffic flow forecasting," pp. 583–593, 2012, doi: 10.1007/s00521-010-0456-7.

[10] P. Poonia and V. K. Jain, "Short-Term Traffic Flow Prediction: Using LSTM," Proc. - 2020 Int. Conf. Emerg. Trends Commun. Control Comput. ICONC3 2020, 2020, doi: 10.1109/ICONC345789.2020.9117329.

[11] T. Ese N et al., "Long Short-Term Memory in Recurrent Neural Networks," vol. 2366, 2366.

[12] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Comput., vol. 12, no. 10, pp. 2451–2471, 2000, doi: 10.1162/089976600300015015.

[13] J. Wang et al., "Spatiotemporal modeling and prediction in cellular networks: A big data-enabled deep learning approach," Proc. - IEEE INFOCOM, 2017, doi: 10.1109/INFOCOM.2017.8057090.

[14] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transp. Res. Part C Emerg. Technol., vol. 54, pp. 187–197, 2015, doi: 10.1016/j.trc.2015.03.014.

[15] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU Neural Network Methods for Traffic

Flow Prediction," no. November 2016, 2018, doi: 10.1109/YAC.2016.7804912.

[16] B. Williams and L. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” Journal of transportation engineering, vol. 129, no. 6, pp. 664– 672, 2003.

[17] E. Zivot and J. Wang, “Vector autoregressive models for multivariate time series,” Modeling Financial Time Series with S-Plus, Springer New York: New York, NY, USA, pp. 385–429, 2006.

[18] R. Chen, C. Liang, W. Hong, and D. Gu, “Forecasting holiday daily tourist flow based on seasonal support vector regression with an adaptive genetic algorithm,” Applied Soft Computing, vol. 26, pp. 435–443, 2015.

[19] U. Johansson, H. Boström, T. Löfström, and H. Linusson, “Regression conformal prediction with random forests,” Machine Learning, vol. 97, no. 1-2, pp. 155–176, 2014.