VIVEKANAND EDUCATION SOCIETY'S

INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Project Report on

Vehicular Traffic Abatement

In partial fulfilment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2017-2018

Submitted by

Pavan Chhatpar (D17-B, 12)

Nimesh Doolani (D17-B, 16)

Aysha Jagiasi (D17-B, 23)

Sumeet Shahani (D17-B, 64)

Project Mentor

Mrs. Priya R. L

(2017-18)

VIVEKANAND EDUCATION SOCIETY'S

INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

<u>Certificate</u>

This is to certify that *Pavan Chhatpar, Nimesh Doolani, Aysha Jagiasi, Sumeet Shahani* of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "***Vehicular Traffic Abatement***" as a part of the coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***<u>Mrs Priya R. L</u>*** in the year 2017-2018.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval

For

B. E (Computer Engineering)

This project report entitled ***Vehicular Traffic Abatement*** by ***Pavan Chhatpar, Nimesh Doolani, Aysha Jagiasi, Sumeet Shahani*** is approved for the degree of B.E, Computer

Internal Examiner	Head of the Department
--	--
External Examiner	Principal
--	--

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--	--
(Signature)	(Signature)
--	--
(Name of student and Roll No.)	(Name of student and Roll No.)
--	--
(Signature)	(Signature)
--	--
(Name of student and Roll No.)	(Name of student and Roll No.)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Priya R.L** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to:-

Course Outcome	**Description of the Course Outcome**
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solution for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

ABSTRACT

Traffic will always be a problem as we move to a modernised way of living. And in those stages of evolution, the complexity increases which in turn demands even more sophisticated measures to tackle them. Machine learning techniques have always helped in making efficient systems with minimal continual human intervention. A similar system is described in this paper. The data was collected from Google Maps. A processing technique was applied to extract features from the raw data. A significant amount of effort was put towards data preparation to make the features suitable for the proposed system. Multiple approaches to model the traffic data were tested before finalizing the model which best suited this proposal. For this research, a few selected areas were chosen. The system built is scalable and can be expanded to cover more areas.

Index

Chapter 1

INTRODUCTION

1.1 Introduction to the project

Traffic is a very important and unavoidable circumstance which can dampen the daily routine and its solutions need to be updated continually. Machine Learning is preferable over deterministic programming for accurate and dynamic solutions and this traffic problem can be solved using this approach. Vehicular Traffic Abatement employs an offline process to predict the best source-to-destination route from a given set of parameters. It is a personalized application that has a wider usability and gets its data sets from Google Maps, which is one of the most trusted API for this purpose.

1.2 Motivation for the project

Traffic is a very important and unavoidable circumstance which can dampen the daily routine and its solutions need to be updated continually. Machine Learning is preferable over deterministic programming for accurate and dynamic solutions and this traffic problem can be solved using this approach.

Existing systems utilise various cameras, sensors, and other hardware which brings in a feasibility factor that needs to be considered. Our system utilities self-learning programs which are highly efficient and cost-effective. Metropolitan areas are always prone to increased vehicular utilisation. Therefore our project finds high practical and widespread use. It can help to solve various issues brought about by traffic and can lead to development of more complex systems.

Vehicular Traffic Abatement employs an offline process to predict the best source-to-destination route from a given set of parameters. It is a personalised application that has a wider usability and gets its data sets from Google Maps, which is one of the most trusted API for this purpose.

1.3 Problem Definition

The difficult challenge of traffic solutions are to manage them without real-time analysis, that is, sans the internet. The proposed idea takes this difficulty into account and revolves around predictions and machine learning. The objective of this project is to provide an offline Android application which will predict real-time traffic across the city, making its users known to alternative routes to their destination. The algorithm will be fed with myriad data sets to train through making the predictions close to accurate.

1.4 Relevance of the Project

Our system uses an offline approach to analyze traffic and update users on its intensity on their routes. Existing systems need an active data connection to synchronize real time traffic to their applications. But this serves as an obstacle for regions with poor to no data connections. This issue is resolved in our project which serves the relevance of the proposed technique.

1.5 Methodology used

1.5.1 Preprocessing

Screenshots taken from Google Maps are processed to remove all colors except those representing traffic. This is followed by sampling of data which provides the location coordinates and the time features of the image processed. The imbalance is resolved using a biasing approach during feature extraction itself.

1.5.2 Machine Learning Algorithms

The algorithms implemented were divided into two sets. The first set was implemented on the unbiased feature extraction. It used K-means clustering and a Deep Neural Network to train over the dataset. The second set was implemented on the biased feature extraction. The algorithms under this approach were: Linear SVM, Bernoulli Naive Bayes, Extra Trees, and Random Forest. A Geohash algorithm was also used alongside these algorithms which basically assigns a hash value to all geolocations in the world.

Chapter 2

LITERATURE SURVEY

2.1 Research Papers

1. **ThammasakThianniwet, SatidchokePhosaard and WasanPattara-Atikom, "Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows", Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009.[1]**
   a. **Abstract--**We proposed a technique to identify road traffic congestion levels from velocity of mobile sensors with high accuracy and consistent with motorists' judgments. The data collection utilized a GPS device, a webcam, and an opinion survey. Human perceptions were used to rate the traffic congestion levels into three levels: light, heavy, and jam. Then the ratings and velocity were fed into a decision tree learning model (J48). We successfully extracted vehicle movement patterns to feed into the learning model using a sliding windows technique. The model achieved accuracy as high as 91.29%. By implementing the model on the existing traffic report systems, the reports will cover on comprehensive areas. The proposed method can be applied to any parts of the world.
   b. **Inference from the paper –**
      i. In this paper, a technique to identify road traffic congestion levels from velocity of mobile sensors is proposed.
      ii. Images of road traffic condition were captured by a video camera mounted on a test vehicle's dashboard. Using these images traffic congestion was categorized into three levels: light, heavy, and jam.
      iii. The velocity extracted was widely fluctuated which was then smoothened out using moving average algorithm.
      iv. Then these traffic congestion levels and velocity were fed into a decision tree learning model and investigated vehicle movement patterns. Vehicle movement patterns were determined and mapped to traffic congestion levels.
      v. These extracted vehicle movement patterns were fed into the learning model using a sliding windows technique. The model achieved accuracy as high as 91.29%.
2. **ArashJahangiri and Hesham A. Rakha, "Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data", IEEE, 2015.[2]**
   a. **Abstract** —This paper adopts different supervised learning methods from the field of machine learning to develop multiclass classifiers that identify the transportation mode, including driving a car, riding a bicycle, riding a bus, walking, and running.

Methods that were considered include K-nearest neighbor, support vector machines (SVMs), and tree-based models that comprise a single decision tree, bagging, and random forest (RF) methods. For training and validating purposes, data were obtained from smartphone sensors, including accelerometer, gyroscope, and rotation vector sensors. K-fold cross-validation as well as out-of-bag error was used for model selection and validation purposes. Several features were created from which a subset was identified through the minimum redundancy maximum relevance method. Data obtained from the smartphone sensors were found to provide important information to distinguish between transportation modes. The performance of different methods was evaluated and compared. The RF and SVM methods were found to produce the best performance. Furthermore, an effort was made to develop a new additional feature that entails creating a combination of other features by adopting a simulated annealing algorithm and a random forest method.

b. **Inference from the paper –**

i. In this paper, a technique to identify road traffic congestion levels from velocity of mobile sensors is proposed

ii. In this paper, Different classifiers were developed using machine learning techniques to identify different transportation modes including bike, car, walk, run, and bus.

iii. In training and testing the classifiers, data were obtained from smartphone sensors such as accelerometer, gyroscope, and rotation vector.

iv. SVM and RF methods were implemented and for each method, parameters that needed to be optimized were examined to conduct a complete model selection.

v. 4-6% of times cars were misclassified as buses.The Random Forest method was found to produce the best overall performance. However, for specific modes (i.e., walk and Run), the SVM outperformed the RF method.

3. **CharlottVallon, ZiyaErcan, Ashwin Carvalho and Francesco Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control", IEEE Intelligent Vehicles Symposium (IV), 2017.[3]**

a. **Abstract** — We study an algorithm that allows a vehicle to autonomously change lanes in a safe but personalized fashion without the driver's explicit initiation (e.g. activating the turn signals). Lane change initiation in autonomous driving is typically based on subjective rules, functions of the positions and relative velocities of surrounding vehicles. This approach is often arbitrary, and not easily adapted to the driving style preferences of an individual driver. Here we propose a data-driven modeling approach to capture the lane change decision behavior of human drivers. We collect data with a test vehicle in typical lane change situations and train classifiers to predict the instant of lane change initiation with respect to the preferences of a particular driver. We integrate this decision logic into a model predictive control (MPC) framework to create a more personalized autonomous lane change experience that satisfies safety and comfort constraints. We show the ability of the decision logic to reproduce and differentiate between two lane changing styles, and demonstrate the safety and effectiveness of the control framework through simulations.

b. **Inference from the paper–**

   i. In this paper, an autonomous lane change algorithm is presented where the lane change decision is determined by a support vector machine (SVM) based classifier.

   ii. Several SVMs are trained using data from actual lane changing and lane keeping demonstrations by human drivers. The performances of the driver is recorded and this data is fed to the SVM classifier which is then trained according to individual driver's driving traits and then performs accordingly.

   iii. The SVM learns whether to continue lane keeping or initiate a lane change, based on the demonstrated preferences of an individual driver.

   iv. Furthermore the lane change execution is adapted to the driving style of particular driver, thereforeresulting in a more personalized driving experience than existing controller.

4. **Hongsuk Yi, HeeJin Jung, Sanghoon Bae, "Deep Neural Networks for Traffic Flow Prediction", IEEE, 2017.[4]**

   a. **Abstract** — Traffic flow prediction is an essential function of traffic information systems. Conventional approaches, using artificial neural networks with narrow network architecture and poor training samples for supervised learning, have been only partially successful. In this paper, a deep-learning neural-network based on TensorFlow™ is suggested for the prediction traffic flow conditions, using real-time traffic data. Until now, no research has applied the TensorFlow™ deep learning neural network model to the estimation of traffic conditions. The suggested supervised model is trained by a deep learning algorithm, which uses real traffic data aggregated every five minutes. Results demonstrate that the model's accuracy rate is around 99%.

b. Inference from the paper

i. In this paper, transportation big data collected from 0.5 million probe vehicles with OBD (car-navigation), was analyzed by Pandas. A DNN model with supervised learning was used to estimate link-based traffic flow conditions using real traffic data. The DNN model was built using TensorFlow from Google Inc., and coded using TFLearn.

ii. It should be noted that it is very important to label the input data when using supervised deep learning DNN.

iii. In this paper, TPI was used to distinguish congested traffic conditions from non-congested traffic conditions, and results show that the suggested 3 layer model could estimate congestion with 99% accuracy.

iv. This researchshows the potential for TensorFlow deep learning models for the accurate analysis of real-time traffic data, and precise estimation of traffic flow conditions.

5. **Jungme Park, Zhihang Chen, Leonidas Kiliaris, Ming L. Kuang, M. AbulMasrur, Anthony M. Phillips, and Yi Lu Murphey, "Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 9, NOVEMBER 2009.[5]**

   a. **Abstract—** Previous research has shown that current driving conditions and driving style have a strong influence over a vehicle's fuel consumption and emissions. This paper presents a methodology for inferring road type and traffic congestion (RT&TC) levels from available onboard vehicle data and then using this information for improved vehicle power management. A machine-learning algorithm has been developed to learn the critical knowledge about fuel efficiency on 11 facility-specific drive cycles representing different road types and traffic congestion levels, as well as a neural learning algorithm for the training of a neural network to predict the RT&TC level. An online University of Michigan-Dearborn intelligent power controller (UMD_IPC) applies this knowledge to real-time vehicle power control to achieve improved fuel efficiency. UMD_IPC has been fully implemented in a conventional (nonhybrid) vehicle model in the powertrain systems analysis toolkit (PSAT) environment. Simulations conducted on the standard drive cycles provided by the PSAT show that the performance of the UMD_IPC algorithm is very close to the offline controller that is generated using a dynamic programming optimization approach. Furthermore, UMD_IPC gives improved fuel consumption in a conventional vehicle, alternating neither the vehicle structure nor its components.

b. Inference from the paper –

i. The authors have presented an intelligent vehicle power controller, namely, the UMD_IPC, which has been developed through machine learning of optimal control parameters with respect to 11 FS road types and traffic congestion levels. UMD_IPC contains a neural network, namely, the NN_RT&TC, designed and trained for in-vehicle prediction of 11 different roadway types and traffic-congestion levels.

ii. This paper also presents a feature extraction algorithm to extract effective features from vehicle speed segments as the input to NN_RT&TC and showed the importance of the two parameters, namely, ΔZ, which is the signal window size, and At, which is the prediction step, with respect to the accuracy of the prediction results of NN_RT&TC.

iii. An offline machine-learning algorithm was proposed to generate a knowledge base that contains the optimal control parameters for all 11 standard FS drive cycles.

6. UrunDogan and Johann Edelbrunner and IoannisIossifidis, "Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior",Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011.[6]

a. Abstract — In the presented work we compare machine learning techniques in the context of lane change behavior performed by humans in a semi-naturalistic simulated environment. We evaluate different learning approaches using differing feature combinations in order to identify appropriate feature, best feature combination, and the most appropriate machine learning technique for the described task. Based on the data acquired from human drivers in the traffic simulator NISYS TRS1, we trained a recurrent neural network, a feed forward neural network and a set of support vector machines. In the followed test drives the system was able to predict lane changes up to 1.5 sec in beforehand.

b. **Inference from the paper –**
    i. In this paper, the lane change paradigm is utilized to investigate the performance and accuracy of machine learning techniques according to the various sensor sources, traffic situations and real human driving data.
    ii. The data was collected from individuals driving in the simulator and performing numerous lane change maneuvers. For each driving episodes, measurable quantities such a velocities, distance to the front vehicle, lateral distance to the lane border etc. were recorded and preprocessed for further analysis.
    iii. Machine learning techniques were used to predict the time of expected lane change maneuvers for straight and curvy road conditions.
    iv. ML approaches like FFNN, RNN, SVM were compared and the comparison showed that SVM leads to the best performance.

7. **Anurag Sharma, Anurendra Kumar, K.V Sameer Raja, ShreeshLadha, "Automatic License Plate Detection", IIT Kanpur, 2015-2016.[7]**
    a. **Abstract**— Automatic License Plate Recognition is an important problem in Computer Vision and Image processing. There are many applications ranging from complex security systems to common areas and from parking admission to traffic control. Automatic license plate recognition has complexity due to diverse effects such as of light and speed of the vehicle. In this project report we explore the methods to detect number plate in a frame using machine learning methods. We first use some basic image processing techniques (which uses some properties of number plate like white background etc.) to filter out possible objects for number plate and then use trained model to detect number plates among them.
    b. **Inference from the paper –**
        i. The focus of this work is on detecting license plate from a video surveillance tape and extracting information from the plate.
        ii. Automatic License plate recognition (ALPR) algorithms are generally composed of the following three steps: 1) extraction of a license plate region; 2) segmentation of the plate characters; and 3) recognition of each character. This task is quite challenging due to the diversity of plate formats and the no uniform outdoor illumination conditions during image acquisition.

iii. Frames are extracted from the videotape, and final data set of 1414 number plates was created. Then SVM classifier, logistic regression and adaboost classifier were used to detect number plate.

8. **Cynthia Jayapal, Sujith Roy. S, "ROAD TRAFFIC CONGESTION MANAGEMENT USING VANET",International Conference on Advances in Human Machine Interaction (HMI - 2016), March 03-05, 2016.[8]**

   a. **Abstract**—With the constant increase in vehicular traffic, existing traffic management solutions have become inefficient. Urbanization has led to an increase in traffic jams and accidents in major cities. In order to accommodate the growing needs of transport systems today, there is a need for an Intelligent Transport System. Vehicular Adhoc Network (VANET) is a growing technology that assists in Intelligent Transport Systems. VANETs enable communication between vehicles as well as fixed infrastructure called Road Side Units (RSU). We propose a distributed, collaborative traffic congestion detection and dissemination system that uses VANET. Each of the driver's smart phones is equipped with a Traffic App which is capable of location detection through Geographic Position based System (GPS). This information is relayed to a remote server which detects traffic congestion. Once congestion is confirmed the congestion information is disseminated to the end user phone through RSUs. The Mobile App transmits the location information at periodic intervals. Using the latitude, longitude and the current time, the location of each vehicle is traced. Using location information, the distance moved by the vehicle at a given time is monitored. If the value is below a fixed threshold, congestion is suspected in a particular area. If many vehicles in the same area send similar messages, traffic congestion is confirmed. Once traffic congestion is confirmed, the vehicles approaching the congested area are informed about the traffic through display boards that are available in the nearest RSUs (traffic signals). The congestion information is also made available through the Mobile App present in vehicles approaching the congested area. The approaching vehicles may take diversion and alleviate congestion. Thus a smart traffic congestion detection and dissemination system is developed to divert incoming vehicles and reduce congestion without human intervention.

   b. **Inference from the paper –**

      i. The steady population increase in urban areas has led to an exponential increase in the number of vehicles on road. Vehicular traffic is one of the most important social and economic issues faced today resulting in congestion.

ii. The vehicular traffic management system consists of a Traffic App installed in the driver's smartphone and a remote server.

iii. The GPS in the driver's smart phone detects the location of the vehicle. The Traffic App extracts this location information and sends it periodically to a remote server. The location information consists of the latitude, longitude, and date and time which can be extracted through a code

iv. Traffic congestion is calculated by considering the location of the vehicle and the number of vehicles in a particular location boundary. If a vehicle is found to transmit the same location information for a prolonged period of time, then the vehicle is marked for traffic boundary.

v. The congestion information provides information about the existence of congestion in a particular area.

2.2 Article referred

1. **Article - Justin Kestelyn, Google Cloud Platform, "Real-time data visualization and machine learning for London traffic analysis", 22 November 2016.[9]**

    a. This article focusses on giving London residents information for making better route decisions during their commutes.

    b. By augmenting real-time data visualization with an ML model, the system can predict areas of congestion during the morning and evening commutes, which currently stand at 30 million daily journeys, and more than 1 million net-new journeys expected by 2018.

    c. The solution uses Google Cloud Platform (GCP) for storage and data processing and provides insights based on 3 months of data from 14,000 traffic sensors across London, amounting to well over 100 billion rows. (From this dataset, 8 days of data from 300 sensors were used for model-training purposes.)

Chapter 3

REQUIREMENT OF PROPOSED SYSTEM

3.1 Functional Requirements

3.1.1 Login

It allows the system to uniquely identify a user. It will help user settings to be kept synchronized across all the devices where the user stays signed in.

3.1.3 Offline Route Generator

To give possible routes to desired destination along with supporting data like time that would be required via each path. The generated routes utilize the traffic predictions

3.1.4 Knowledge Update

For users' app to regularly update knowledge for prediction of traffic in future offline scenarios, the app downloads the trained ML models when connected to internet.

3.1.5 Training Center

For training the algorithm on server with new data sets time-to-time ensuring a good accuracy and up-to-date information about traffic patterns.

3.1.6 Traffic Data Extraction

To extract traffic data set from screenshots acquired from Google Maps using Image Processing techniques. This also generates the input training set in the format that the training algorithm demands.

3.1.7 Data Service Layer

To make server's data set available to client apps. It also provides user authentication and access control by exposing limited amount of data for which the user is authorized to access.

3.2 Non-Functional Requirements

3.2.1 Offline Availability

For users to be able to get best routes based on traffic condition predictions.

3.2.2 Non-intrusive Route Prediction

The routing process runs in background without hindering the user due to stuttering UI during otherwise heavy foreground tasks.

3.2.3 Reliability

To ensure correct traffic prediction authentic sources of data are used.

3.2.4 Low Power Consumption

Without the necessity of a constant internet connection, offline apps reduce the load on device's hardware.

3.3 Constraints

3.3.1 Update Constraints

VTA does not take in account immediate events that may change the traffic conditions of an area temporarily; these include accidents or natural calamities which may block certain roads temporarily. VTA will not be able to cater to these problems by instantly updating routing mechanisms. In order to have the most up-to-date traffic pattern information, the client application needs to get updated weights from server as and when the server updates them.

3.3.2 Algorithm Constraints

Prediction accuracy depends on the data set and the ML algorithm used for training. The data set should be adequate enough to provide knowledge about traffic patterns at all times of the day. To make the application work offline, the prediction and route generation calculations happen on the client's device.

3.4 Hardware, Software, Technology and Tools - Requirements

3.4.1 Hardware:

Smart-phones with on device GPS module

3.4.2 Software:

Linux Server, Android OS

3.4.3 Technology:

- **Image Processing:**

  The screenshots obtained from Google Maps are processed to filter out traffic information of streets and to link this information with the geo-location, time of day and road type.

- **Machine Learning Techniques**

  Supervised and unsupervised ML techniques are used for predictive reports on traffic, this is the core algorithm of the system.

3.4.4 Tools utilized for the proposed system

- **API:**

  TensorFlow, Node.js, sklearn predictor

- **Softwares& IDE:**

  Pycharm, Visual Studio Code, Android Studio

3.5 Selection of Hardware, Software, Technology and Tools

3.5.1 Hardware

Android phone in order to run the application. The GPS of the phone should be enabled to locate the user's current location

3.5.2 Software

Android OS to test the application

3.5.3 Technology

Image Processing is used to extract the traffic information from the screenshots of the Google Maps, ML techniques are used to predict the traffic on the routes.

3.5.4 Tools

Node.js to enable JavaScript processing on the server side

TensorFlow and sklearn Predictor to implement the ML algorithms

Softwares and IDEs to develop the application with ease

Chapter 4

PROPOSED DESIGN

4.1 Block diagram representation of the proposed system

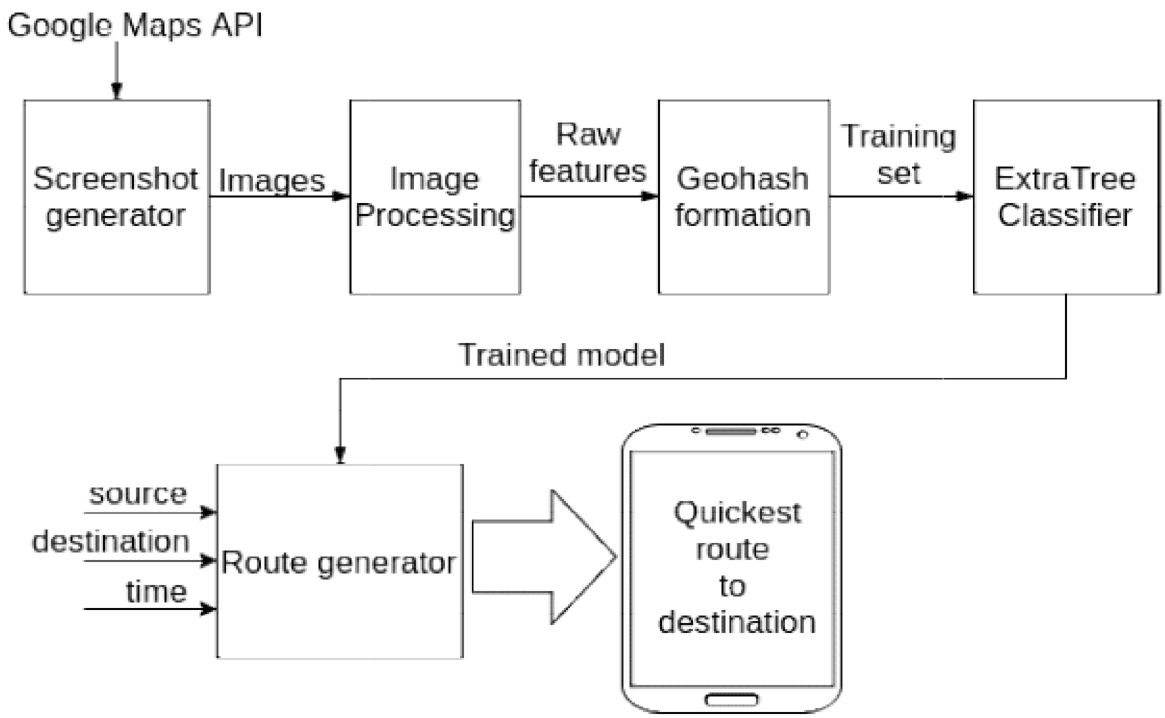


Figure 1: Block Diagram

In the Block Diagram, there is one external interface, Google Maps API, and four modules, Screenshot generator, Image Processing, Geohash formation and ExtraTree Classifier training which works on server side. The results of these are stored on a database will be later used by client application. Route generator takes user input through the application and gives the best route using the trained models' predictions. It takes source and destination addresses, records journey time and splits up alternative routes into segments for calculating prediction

4.2 Modular Diagram

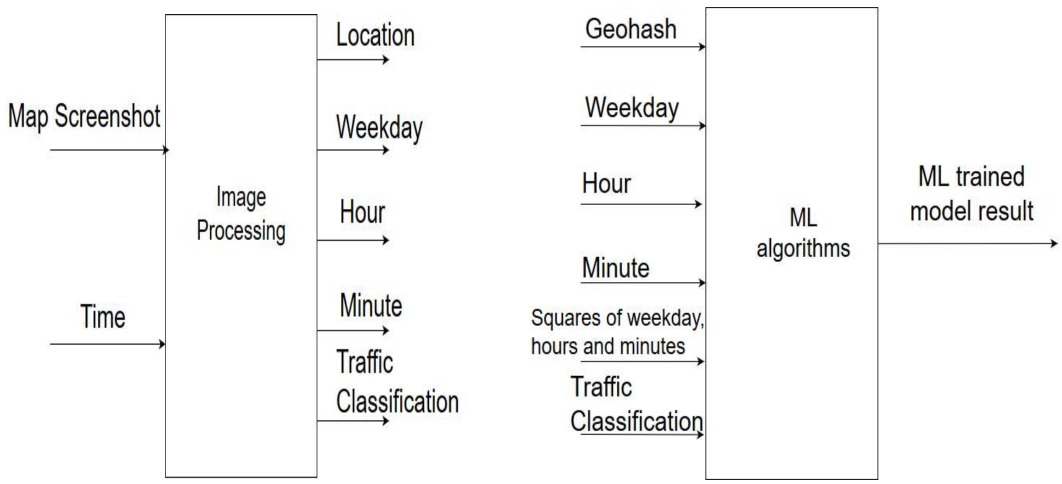


Figure 2: Modular Representation

4.3 Design of Proposed System

4.3.1 Data Flow Diagram

Level 0:

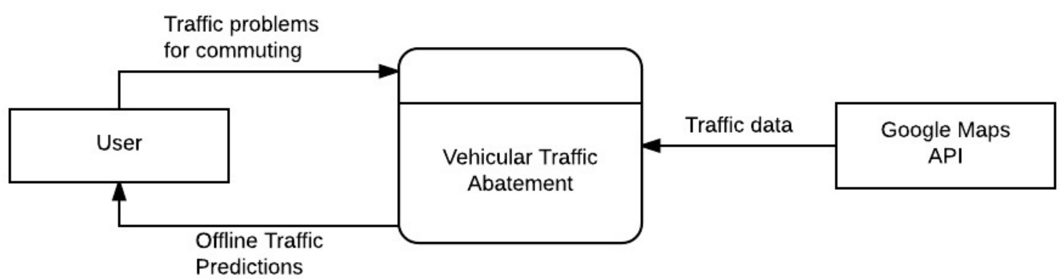


Figure 3: DFD Level 0

Level 1:

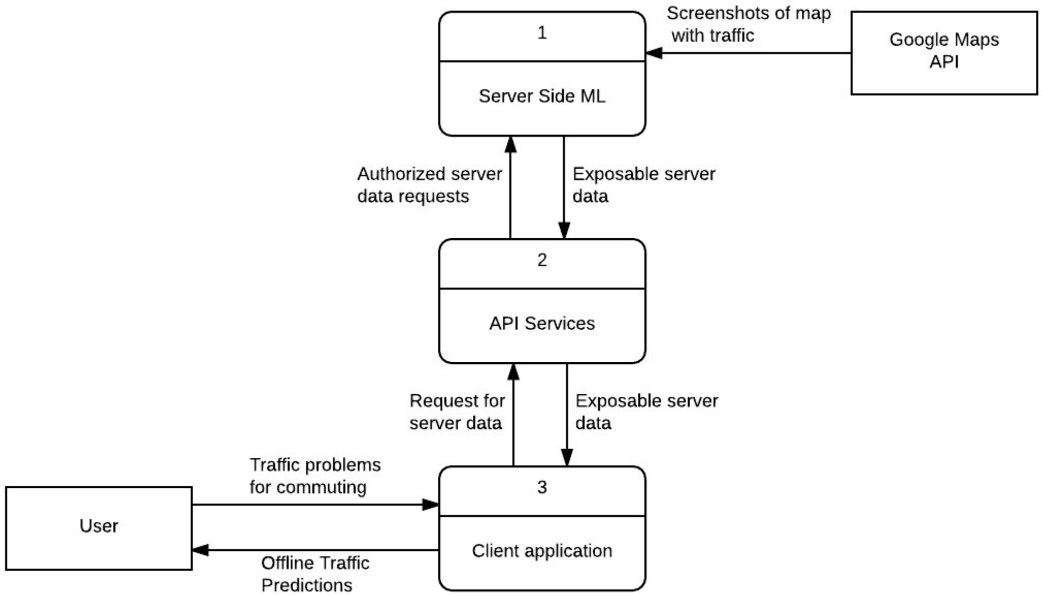


Figure 4: DFD Level 1

Level 2:

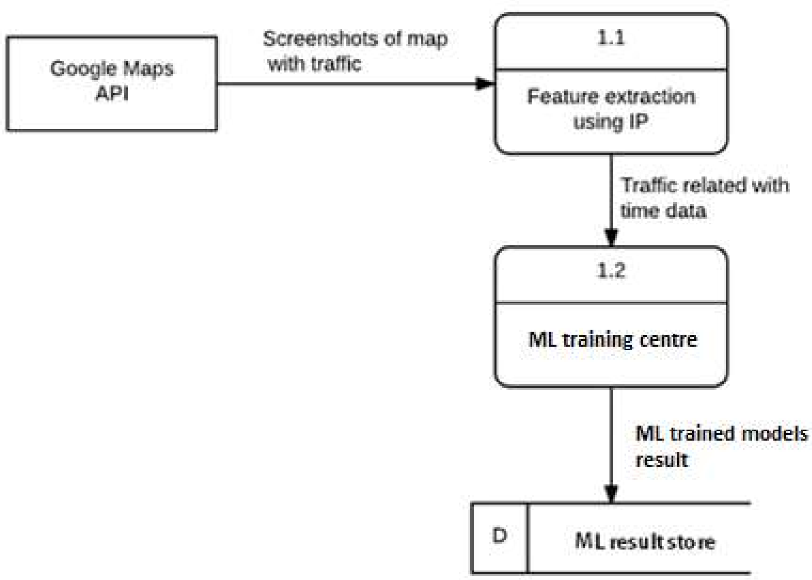


Figure 5: DFD Level 2 (1)

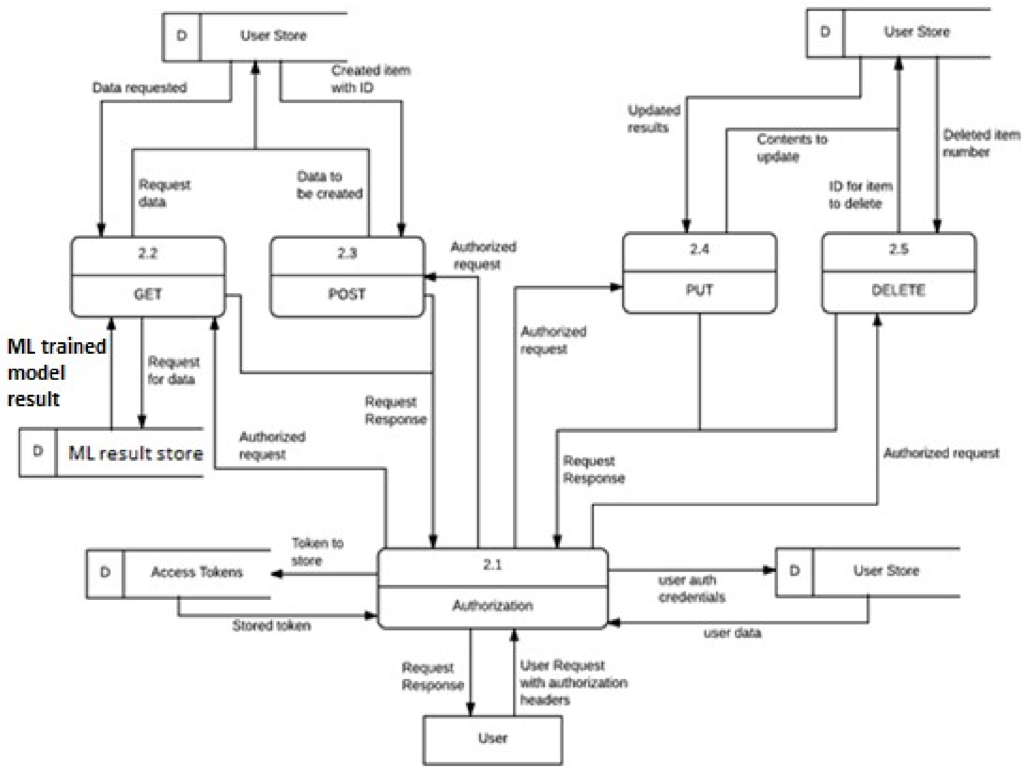


Figure 6: DFD Level 2 (2)

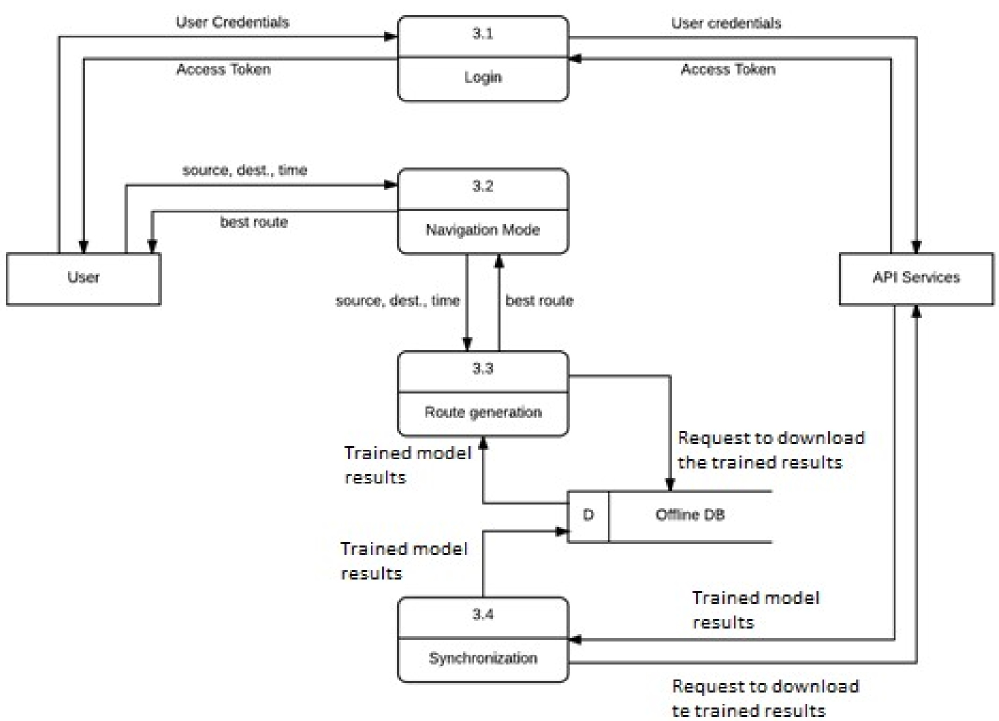


Figure 7: DFD Level 2 (3)

Level 3:

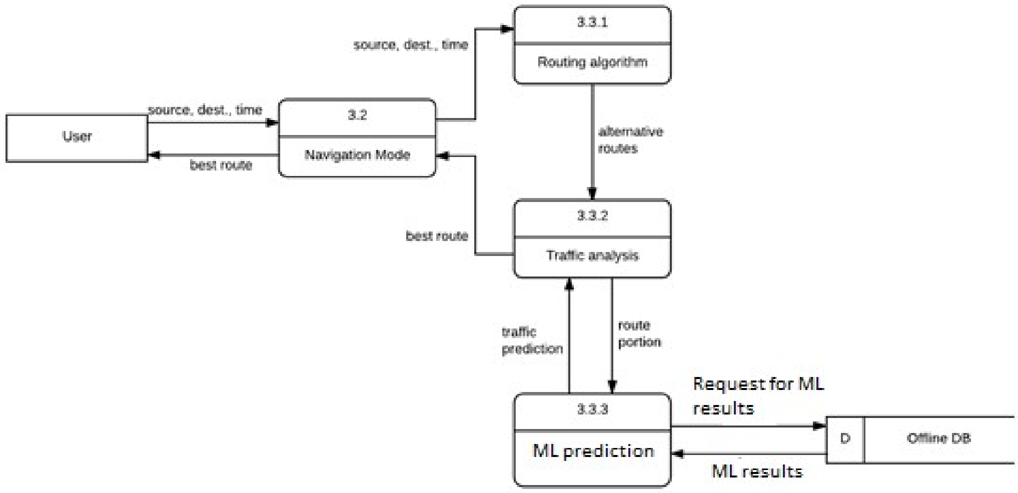


Figure 8: DFD Level 3

4.4.2 Flowchart

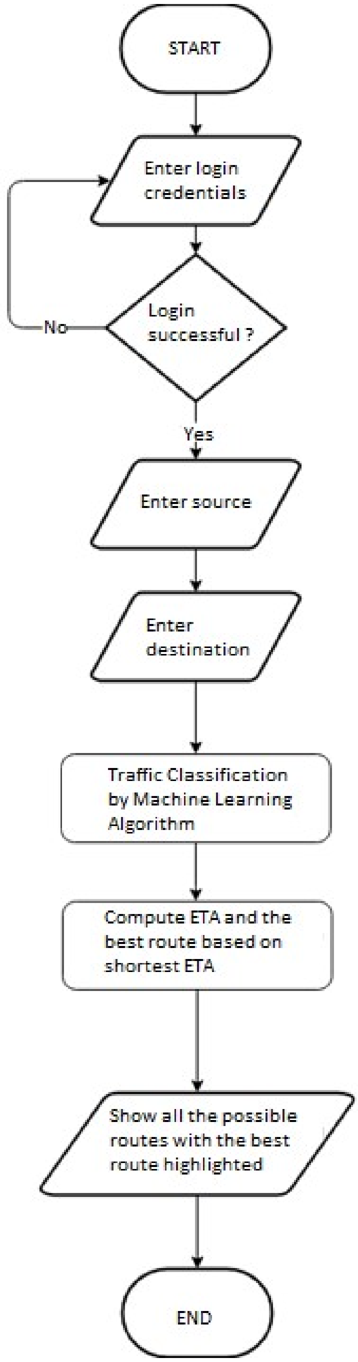


Figure 9: Flowchart

4.4.3 Activity Diagram

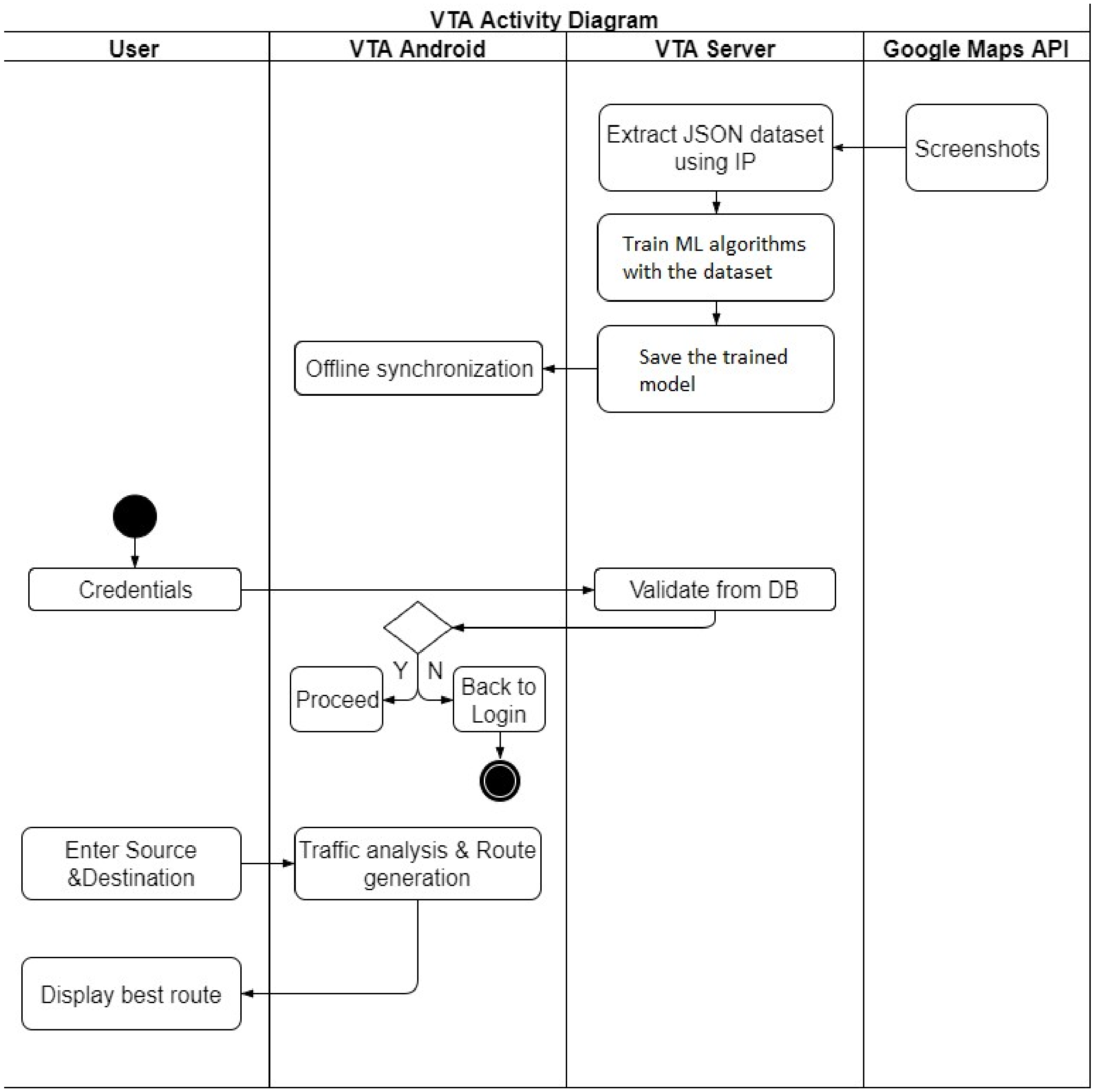


Figure 10: Activity Diagram

4.4.4 ER Diagram

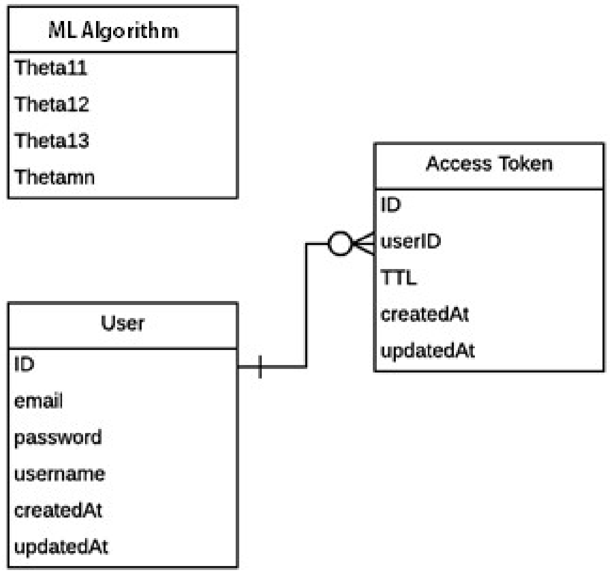


Figure 11: ER Diagram

4.5 Project Scheduling & Tracking using Timeline / Gantt chart

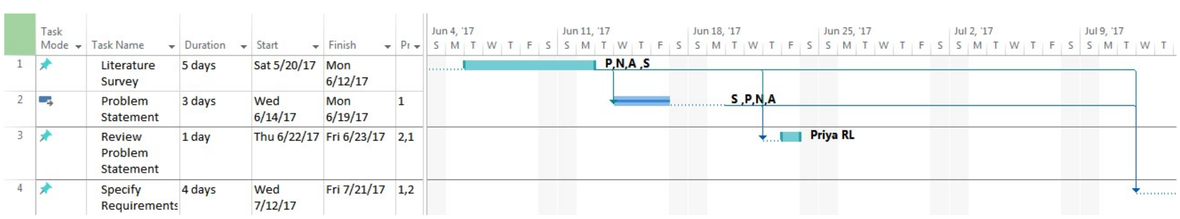


Figure 12: Gantt chart (1)

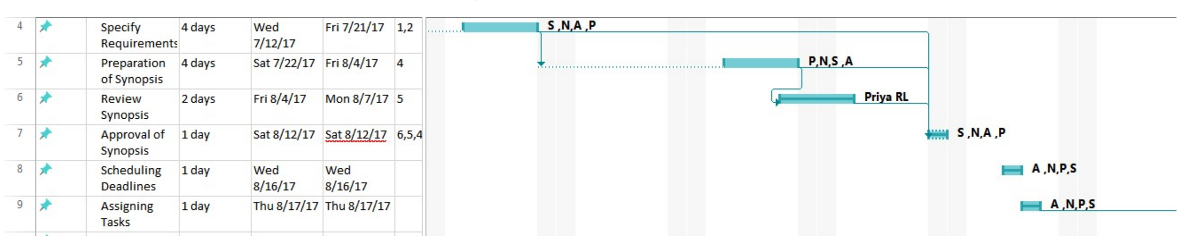


Figure 13: Gantt chart (2)

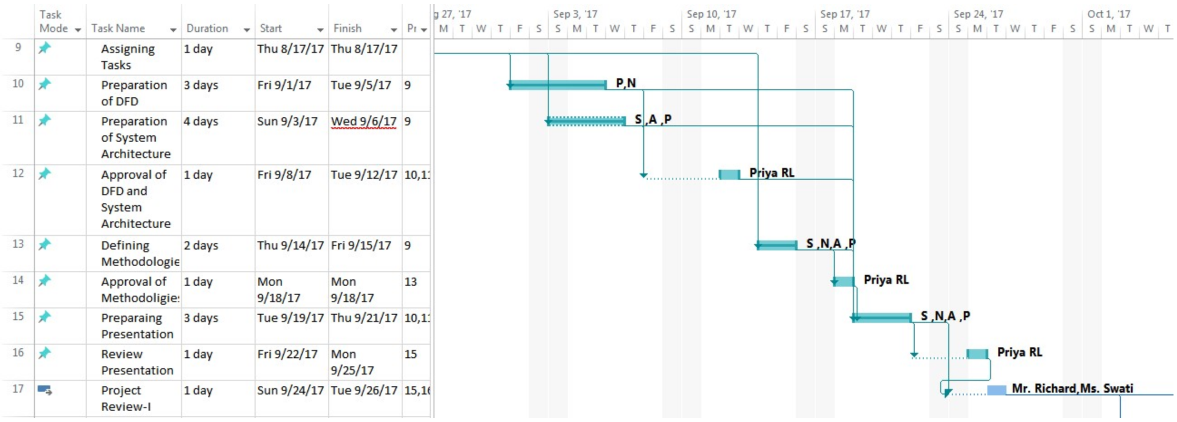


Figure 14: Gantt chart (3)

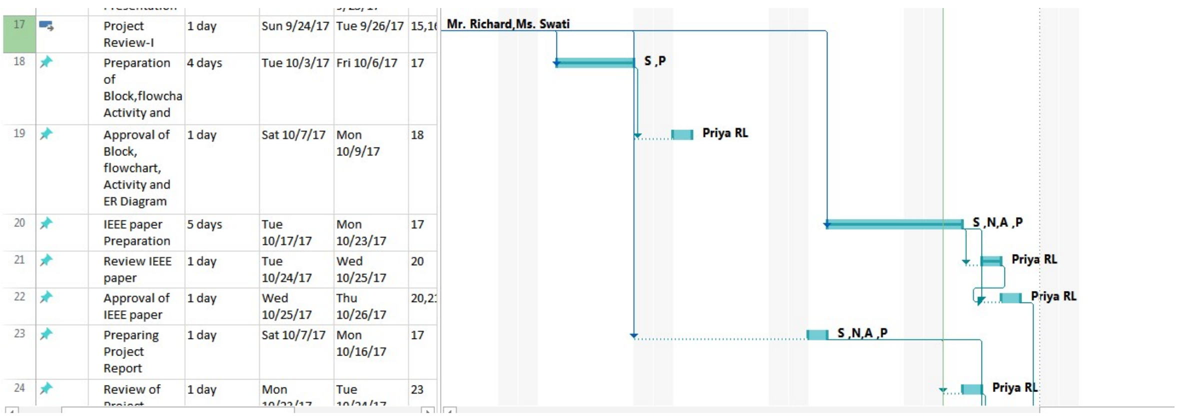


Figure 15: Gantt chart (4)

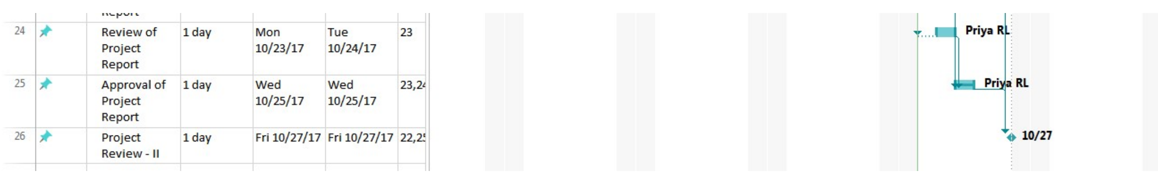


Figure 16: Gantt chart (5)

Chapter 5

IMPLEMENTATION DETAILS

5.1 Algorithms for the respective modules developed

1. kmeans+neural net classification
2. geohash+Linear SVM
3. geohash+BernoulliNB
4. geohash+ExtraTreesClassifier
5. geohash+RandomForestClassifier

5.1.1 Image Processing

- Image Processing for Algorithm 1
  - From the screenshots obtained from Google maps, latitude and longitude along with traffic data of a location are extracted. Data from a new location is collected only if there is no other existing sample in a 50m radius with the same target traffic. Screenshot is captured every 15 minutes.
  - This creates a highly imbalanced dataset with major class being 61% (Green) of the dataset and minor class just 3% (Dark red).

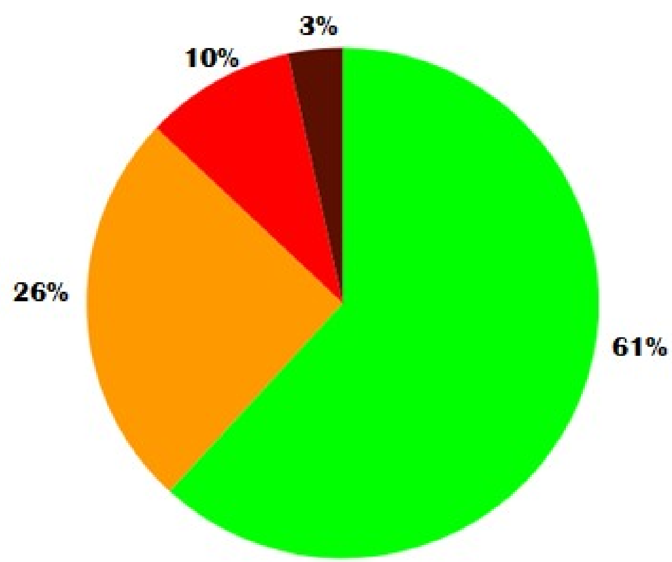


Figure 17: Initial class distribution for algorithm 1

  - The figure shows imbalance distribution of target traffic. To solve this problem, the minor class was discarded and two major classes were reduced to the size of the third class.

- This reduction was achieved by clustering the data of each of the two major classes into the same number of clusters as the size of the third class. Clustering was achieved by K-Means which is an unsupervised Machine Learning algorithm with an ability to cluster data into a said number of clusters.

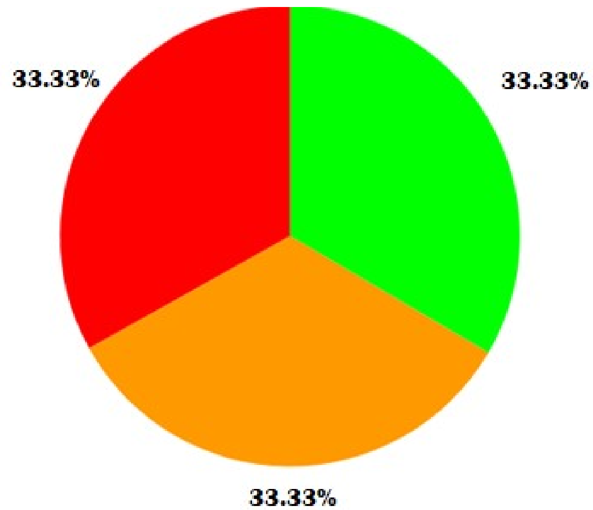


Figure 18: Balanced class distribution for algorithm 1

- Image Processing for Algorithms 2-5
  - Scan through the filtered versions of each image for each class and sample data at every 100 meters for green, 60 meters for orange, 20 meters for red and 10 meters for dark red. This is done to solve the imbalance of the target traffic.
  - Solves imbalance issue to some extent but does not eliminate it entirely.

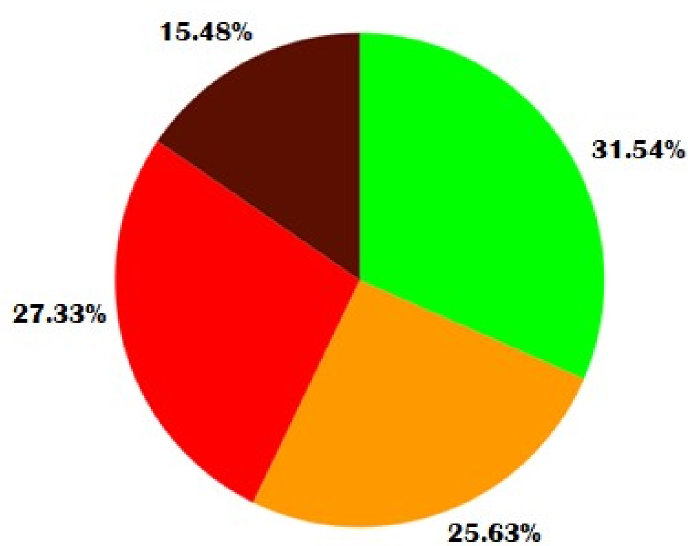


Figure 19: Class distribution for algorithms 2-5

5.1.2 Machine Learning Algorithms

- Algorithm 1 (Deep Neural Network)
  - A deep neural network was used to train over the dataset. The data was normalized in the range of [-1, 1] which is known to be well suited for neural networks.
  - The model had 3 hidden layers with 10 neurons each, using the ReLU activation function. The learning rate was set to 0.1. Training happened using 100 complete epochs over mini batches of 1/8th of the total dataset, giving a total of 800 iterations.
  - This algorithm was used for optimization of the loss function for our model.
  - Tool used - TensorFlow
- Algorithm 2 (Linear SVM)
  - Balancing is applied using class weights to penalize loss function
  - Data divided over 8 models considering 3 hours each
  - Suffers from under fitting as model complexity is too less in comparison with number of samples
  - Tool used – sklearn
- Algorithm 3 (Bernoulli Naive Bayes)
  - Balancing is applied using class weights to penalize loss function
  - Data divided over 8 models considering 3 hours each
  - Distorts the class distribution of dataset immensely biasing it to favour minor class a lot.
  - Tool used – sklearn
- Algorithm 4 (Extra Trees Classifier)
  - Balancing is applied using class weights to penalize loss function
  - Data divided over 8 models considering 3 hours each
  - Over fitting tendency is lesser than RFC inherently and further restricted by limiting depth of tree
  - Tool used – sklearn
- Algorithm 5 (Random Forest Classifier)
  - Balancing is applied using class weights to penalize loss function
  - Data divided over 8 models considering 3 hours each
  - Tends to overfit due to bootstrapping concept
  - An effort to limit overfitting is applied by restricting depth of tree
  - Tool used – sklearn

5.2 Comparative analysis with the existing Algorithms

Existing System	VTA
In [1], Captured Images of road traffic by a video camera mounted on a test vehicle's dashboard. To categories traffic intensity Sliding window protocol	Traffic intensity classified by color densities on the traffic map screenshots. No sliding window.
In [11], Consists of Hierarchy of agents in the auctioning agent control system.	Our system does not handle and alter the traffic system. It calculates the best possible path from a source to a destination. The junction parameter factors in the algorithm on the physical end of a user's device.
In [2], Data was obtained from smartphone sensors such as accelerometer, gyroscope, and rotation vector which were found to have important information for the purpose of mode recognition.	Our system uses the Google API which already recognizes and provides us with the different modes. For positioning, specifically GPS sensor is used to locate the user.
In [9], Google Cloud Platform for storage and processing of data from 14,000 traffic sensors across London.	Our system does not deploy any sensors to gather data sets. Data extraction from the API and simply routing. A user may not travel at any given instant.
In [10], The data was from a fleet of about 500 taxis in San Francisco. Each taxi provides a measurement of its location approximately once every minute	The data is collected from google maps. No help was taken from other resource to collect the data.

(generally between 40 and 100 seconds)	
In addition to its location, the taxi also reports whether or not it is carrying a customer.	No such information is obtained through Google maps
The time interval to collect the data from the drivers is 30 minutes.	Screenshot of the Google maps is taken every 15 minute to capture the traffic information
Average speed from each observation is computed and assigned to the path.	Speed of the path is based on the traffic and the road-limit.
Graphical and baseline models were used for training purpose wherein Baseline proved to be the better one.	kmeans+neural net classification, Linear SVM, BernoulliNB, ExtraTreesClassifier, RandomForestClassifier models were used. ExtraTreesClassifier proved to be the best.
The baseline approach resulted in 40% error.	The ExtraTreesClassifier results in approx. 9% error rate.

Table 1: Comparison between existing and proposed system

5.3 Evaluation of the developed System

Performance of each model, post the training, was evaluated on an unseen test set. The Figure 20 shows the accuracies of the ML models. Geohash-ExtraTrees resulted the best when predicting the traffic.

Another evaluation of the algorithms was done by comparing the distribution of target classes in the test set with that of the predicted results, as shown in table 2. A general observation would be to get almost the same distribution in both cases; so the difference between ratio of actual presence of a class and ratio of predicted presence of that class should be zero or as close to zero as possible.

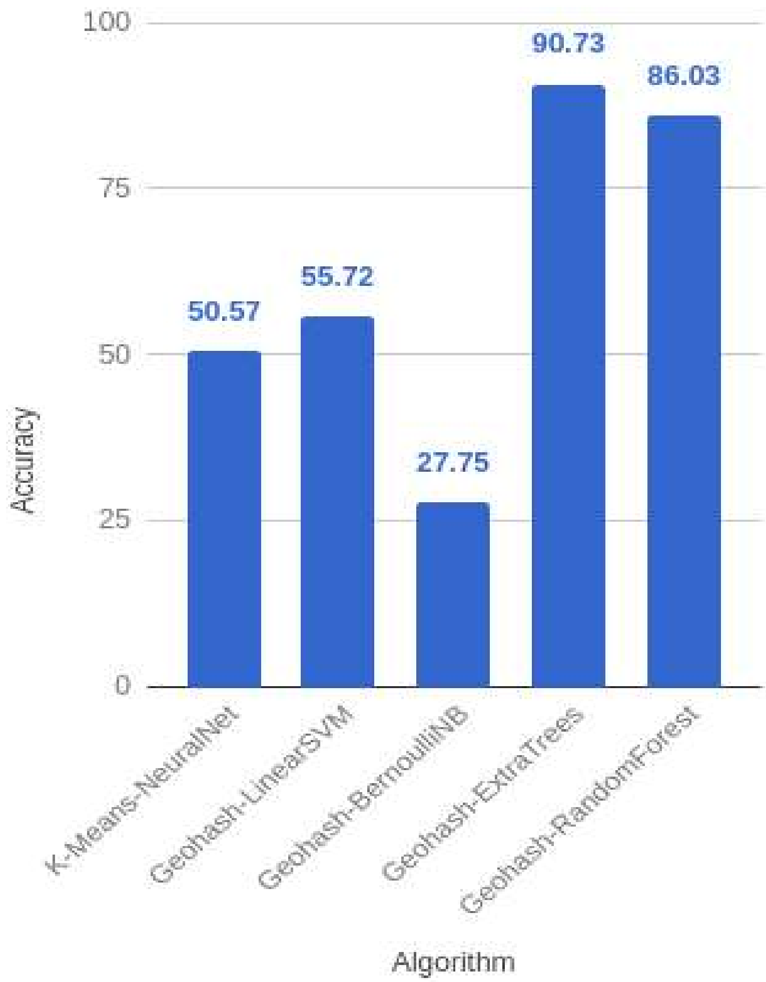


Figure 20: Comparing accuracies of alternative algorithms

Algorithm	Green	Orange	Red	Dark red
K-Means-NeuralNet	-5.32	-18.73	24.05	0
Geohash-LinearSVM	20.18	-16.75	-4.52	1.09
Geohash-BernoulliNB	-24.64	-19.94	-7.31	51.89
Geohash-ExtraTrees	-0.21	-0.05	0.03	0.23
Geohash-RandomForest	1.43	-0.66	-0.55	-0.22

Table 2: Difference in expected and predicted distribution of output classes

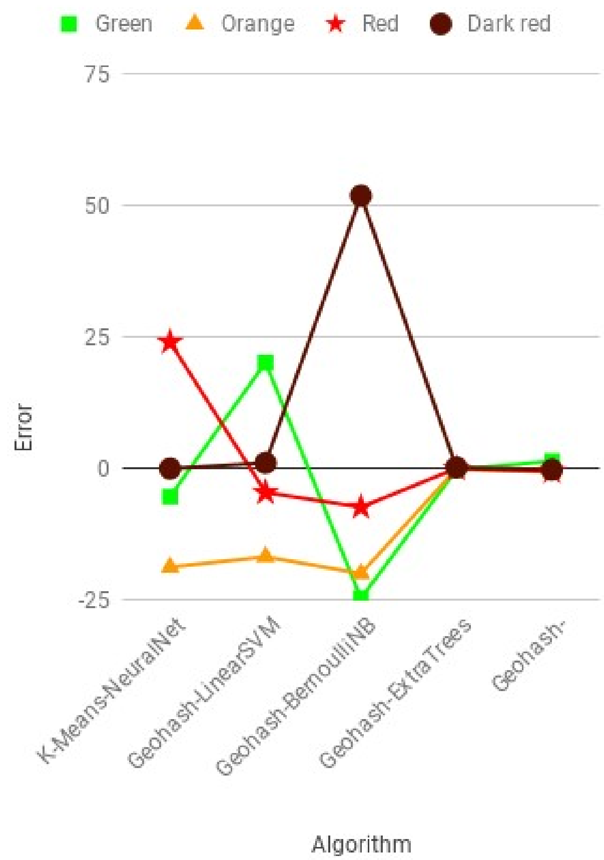


Figure 21: Graphical view of difference in expected and predicted distribution of output classes

The Figure 26 shows the error in the prediction result obtained from the trained models. It can be clearly seen that Geohash-ExtraTrees predicts the all the classes with negligible error. Hence, it proves to be the best algorithm for the proposed system

Chapter 6

TESTING

6.1 Unit Testing

In this stage of testing, the individual units of the application are tested. The errors reported during the test are fixed and the testing is performed recursively until all the errors are fixed and the application gives desired output as mentioned in the requirements.

The following table shows the various test cases under unit testing:

Test Case No.	Test Case	Description	Input	Expected Output	Actual Output	Pass/Fail
1.	Signup	Validating Email address and Password.	Email, password, confirm password	Email and password entered are valid	Email and password entered are valid	Pass
2.	Geohash formation	On entering latitude and longitude of a location, geohash method should form a 7 digit geohash	Latitude, longitude	Geohash string of length 7	Geohash string of length 7	Pass
3.	ETA	Calculate ETA to reach to a location	Source, destination, traffic, speed	Calculate the estimate time to arrive to the location	Calculates the estimate time to arrive to the location	Pass

Table 3: Unit test cases

6.2 Integration Testing

Integration testing is done using Travis Continuous Integration (CI) service. It is used to build and test the projects hosted on GitHub. It is configured by adding a file named .travis.ymlto the root directory of the repository. This file specifies the programming language used, the desired building and testing environment and various other parameters.

When Travis CI has been activated for a given repository, GitHub will notify it whenever new commits are pushed to that repository or a pull request is submitted. It can also be configured to only run for specific branches. Travis CI will then check out the relevant branch and run the commands specified in .travis.yml, which usually build the software and run any automated tests. The build is then run and if one or more of the tasks fails, the build is considered broken. If none of the tasks fail, the build is considered passed.

6.3 User Acceptance Testing

The android application is tested on multiple users of different ages to understand the user acceptance of the application. Within a few minutes, the users were able to use to application easily. With a simple UI design similar to the Gmaps, the users were easily able to understand the purpose and the usage of the application. The users were able to explore the features of the app by themselves.

Chapter 7

RESULT ANALYSIS

7.1 Simulation Model

An android application is developed to test whether the trained model gives accurate results or not. The application starts by displaying the map to the user. The user is required to enter the desired source and destination of the journey. The sourceor destination could be easily out of the area and the algorithm would not give sound inference. This is avoided by geofencing the supported areas. Any location found outside the considered region would simply trigger a notification to the user prompting them that location is out of service. This also features into the routing algorithm. Any route that took the transit from outside the region would simply be discarded. A location picker is provided for the user to pick the source and destination location from the UI itself. Once the source and destination are entered by the user, the application loads the traffic data from the trained model and the user is shown all the possible routes from source to destination along with the traffic information. The application also calculates ETA of all the routes and the route with the shortest ETA is highlighted. ETA and shortest distance of the best route is displayed on the screen. Through regular updates of re-trained model, the client applications have the most up to date information about the city traffic patterns

7.2 Parameters Considered

7.2.1 Parameters considered for Algorithm 1:

- **Latitude and Longitude**

  Latitude and Longitude are obtained via shift of origin and scaling applied over pixel coordinates of the image. Latitude and Longitude pair is considered only if there is no other existing sample in a 50m radius with the same target traffic.

- **Time**

  Weekday, Hour and Minute are obtained from creation time of the screenshot.

- **Traffic Classification (Target Value)**

  The target value is one of the four classes, light traffic, medium traffic, heavy traffic and very heavy traffic, based on the corresponding color shown on that coordinate.

7.2.2 Parameters considered for Algorithms 2-5:

- **Geohash**

  Latitude and Longitude pair is transformed into Geohash which is string of length 7.The Geohash Algorithm assigns a hash value to all geolocations in the world and its precision is adjustable

- **Time**

  Weekday, Hour and Minute are obtained from creation time of the screenshot. To obtain high accuracy, extra input features such as square, square root, cube and cube root of the features were considered as input parameters. Highest accuracy was obtained when square of these features was used.

- **Traffic Classification (Target Value)**

  The target value is one of the four classes, light traffic, medium traffic, heavy traffic and very heavy traffic, based on the corresponding color shown on that coordinate.

7.3 Screenshots of the User Interface.

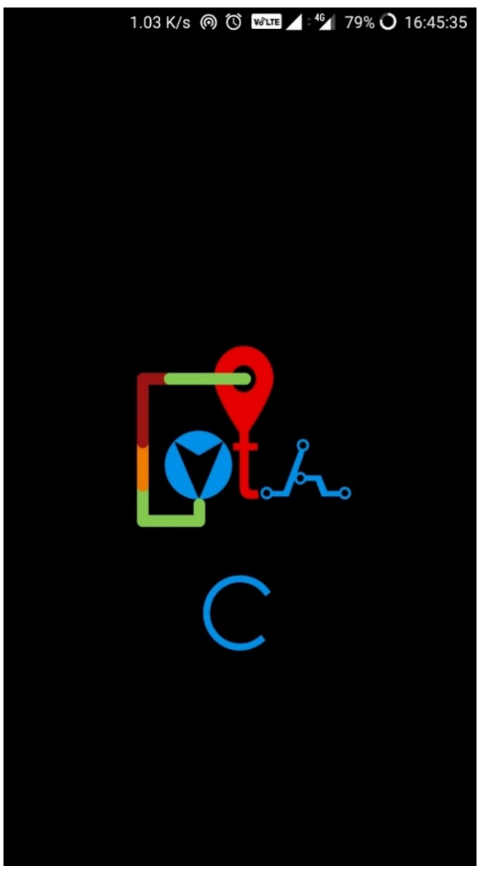


Figure 22: Splash Screen

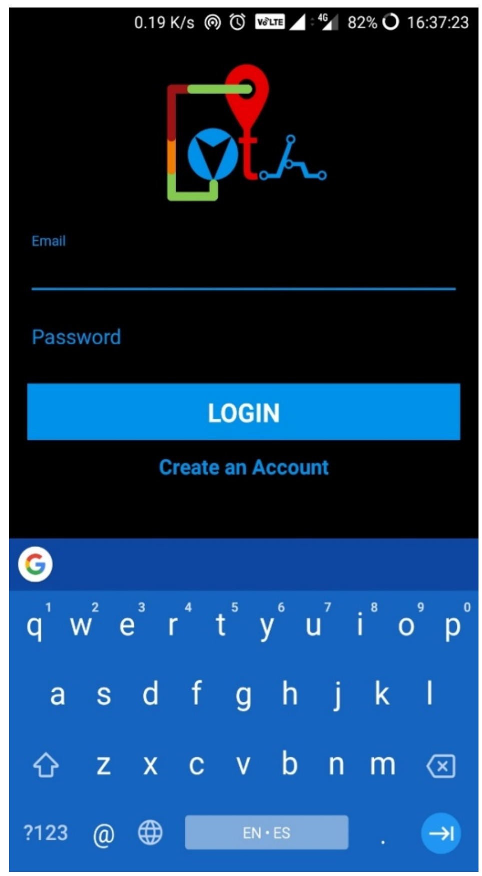


Figure 23: Login Screen

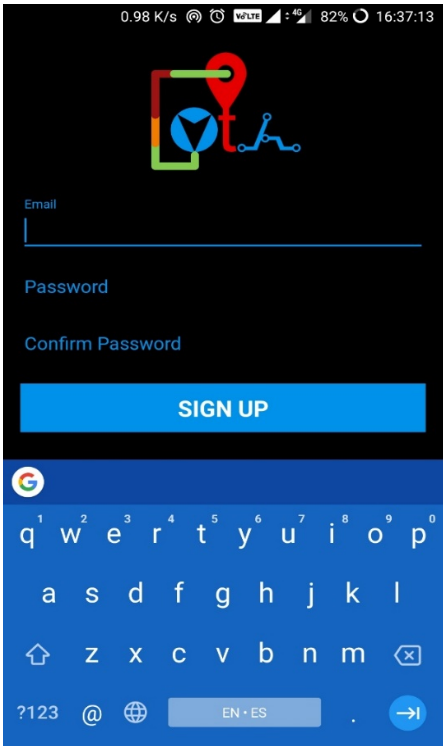


Figure 24: Signup Screen

Figure 25: Home Screen

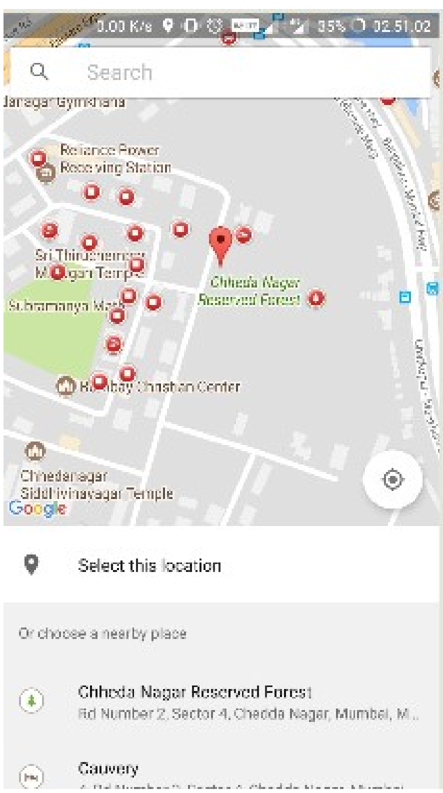


Figure 26: Location Picker

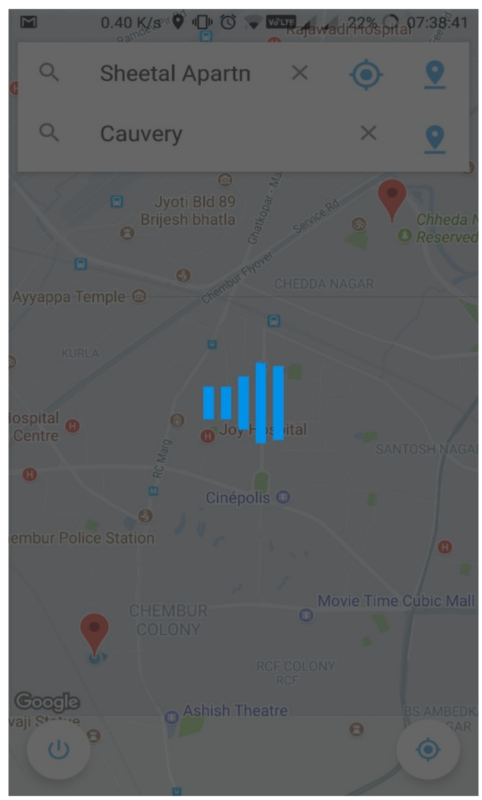


Figure 27: Loading the route from source to destination

7.4 Graphical Outputs

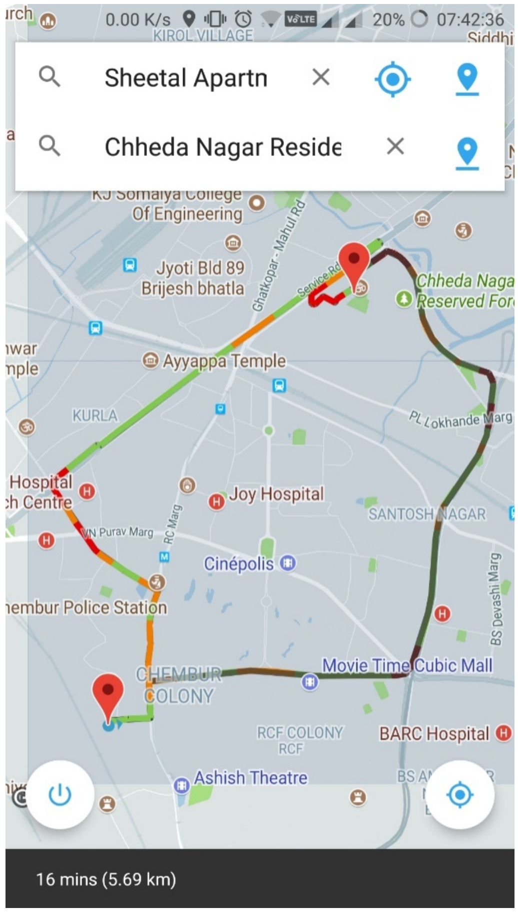


Figure 28: Example route generation based on predicted traffic

All the routes from source and destination are displayed on the screen and the best route (the one with the shortest ETA) is highlighted. Simultaneously, the distance and ETA of the best route is printed on the screen.

7.5 Reports Generated

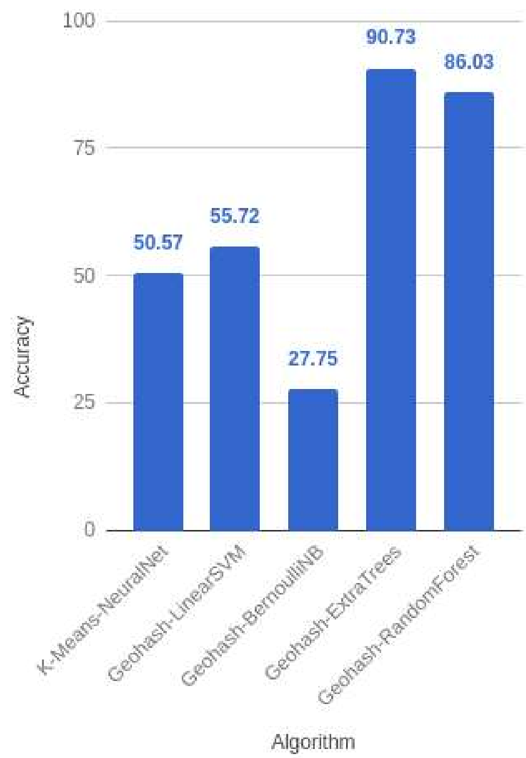


Figure 29: Comparing accuracies of alternative algorithms

In the figure, Accuracy obtained from all the training models is shown in the form of a bar graph. The accuracies were calculated as the ratio of correct predictions with total predictions. Highest accuracy was obtained from Geohash-ExtraTrees classifier.

ETA

Distance between the source and destination is calculated. The four traffic classifications are assigned with some speeds. The distance and the speed pairs between the two points can help us arrive to an average time required to travel from one point to other. This is done for all the possible routes between source and destination and the route with the shortest Estimated Time of Arrival (ETA) is highlighted. The distance and the ETA of the best route is displayed on the screen.

Chapter 8

CONCLUSION

8.1 Limitations

- VTA does not take in account immediate events that may change the traffic conditions of an area temporarily
- The phone needs to sync the trained models on a timely basis, which would require internet connection
- Prediction accuracy depends on the data set
- Client app requires to do the prediction calculations.

8.2 Conclusion

- In this study, we investigated alternative techniques to automatically classify the road traffic congestion levels by training our system to predict traffic information.
- The technique minimally requires the colour codes of traffic from gmaps, time and latitude and longitude which can be collected from screenshot of a particular area from gmaps.
- The screenshots can be taken every 15 minutes to sample real time traffic accurately.
- Also, the route to be taken for the journey will be selected based on predictive results of the trained algorithm.

8.3 Future Scope

- The implementation of application is targeted on the android OS. In future, other prominent mobile OS can be taken into consideration to expand the solution's reach.
- In future, the app can be modified to suggest the user to opt for public or private transport based on the traffic on the routes between sources to destination.
- The area of service of the application can be expanded to cover a whole city.

REFERENCES

References

[1] Thammasak Thianniwet, Satidchoke Phosaard and Wasan Pattara-Atikom, “Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows”, Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009.

[2] Arash Jahangiri and Hesham A. Rakha, “Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data”, IEEE, 2015.

[3] Charlott Vallon, Ziya Ercan, Ashwin Carvalho and Francesco Borrelli, “A machine learning approach for personalized autonomous lane change initiation and control”, IEEE Intelligent Vehicles Symposium (IV), 2017.

[4] Hongsuk Yi, HeeJin Jung, Sanghoon Bae, “Deep Neural Networks for Traffic Flow Prediction”, IEEE, 2017.

[5] Jungme Park, Zhihang Chen, Leonidas Kiliaris, Ming L. Kuang, M. Abul Masrur, Anthony M. Phillips, and Yi Lu Murphey, “Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 9, NOVEMBER 2009

[6] Urun Dogan and Johann Edel brunner and Ioannis Iossifidis, “Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior”, Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011.

[7] Anurag Sharma, Anurendra Kumar, K.V Sameer Raja, Shreesh Ladha, “Automatic License Plate Detection”, IIT Kanpur, 2015-2016

[8] Cynthia Jayapal, Sujith Roy. S, “ROAD TRAFFIC CONGESTION MANAGEMENT USING VANET”,International Conference on Advances in Human Machine Interaction (HMI - 2016), March 03-05, 2016.

[9] Justin Kestelyn, Google Cloud Platform, “Real-time data visualization and machine learning for London traffic analysis”, 22 November 2016.

[10] Ryan Jay Herring, “Real-Time Trac Modeling and Estimation with Streaming Probe Data using Machine Learning”, Fall 2010

[11] Simon Box and Ben Waterson, “An automated signalized junction controller that learns strategies by temporal difference reinforcement learning”, Engineering Applications of Artificial Intelligence, March 9, 2012

APPENDIX

10.1 List of Figures

10.2 List of Tables

10.3 Paper Publications

10.3.1 Publication details of the paper-I

Conference: International Conference on Smart City and Emerging Technologies, 2018

Paper Title: Machine Learning Solutions to Vehicular Traffic Congestion

Presented By: Pavan Chhatpar, Sumeet Shahani

Authors: Pavan Chhatpar, Nimesh Doolani, Sumeet Shahani, and Priya R. L

10/31/2017 Welcome to Vivekanand Education Society Mail - Acceptance of Abstract submitted by you

PAVAN CHHATPAR <pavan.chhatpar@ves.ac.in>

Acceptance of Abstract submitted by you
1 message

Universal Education <noreply@universal.edu.in> Fri, Oct 27, 2017 at 2:27 PM
To: pavan.chhatpar@ves.ac.in

Dear PAVAN CHHATPAR and Mrs. Priya R.L,

Congratulations. Your Paper code RoWL6q bearing Title Machine Learning Solutions to Vehicular Traffic has been accepted by our reviewers.

You are requested to submit your full-length paper in the prescribed format (details available in the attached file) in ConferenceManager as an attachment, within 10 days.

Our Best,
ICSCET – Conference Chair

10.3.2 Paper I

Machine Learning Solutions to Vehicular Traffic Congestion

Mrs. Priya R.L, Assistant Professor, Computer Engg., VESIT, Pavan Chhatpar, Computer Engg., VESIT, Nimesh Doolani, Computer Engg., VESIT and Sumeet Shahani, Computer Engg., VESIT

***Abstract*--** **Traffic management of metropolitan cities in India is becoming a challenging factor day by day. Efficiency of existing traffic management solutions is decreasing, as the number of private vehicles is on the rise. In the context of increasing complexity of urban traffic, the work proposes an android application that makes use of real-time traffic data and predicts the traffic densities of entire map area in offline mode. The bigger picture here is reduction of congested roads all over the city. This mechanism will help to minimize the battery consumption of mobile devices. It provides predictive analysis of traffic in a given area using machine learning techniques like Neural Networks. It also specifically suggests best routes from source to destination based on the traffic data.**

I. Introduction

Bringing autonomy in systems that assist humans in everyday tasks and adapt with their lifestyle is one of the grand challenges in modern computer science. Autonomous route prediction systems which can help with efficient course from a source to a destination are definite paths toward convenient transits.

While a variety of systems make use of cameras, traffic sensors, and other state-of-the-art hardwares, self-learning algorithms provide an effective and cost-efficient approach to solving various vehicular problems. This tends to decrease the load on traffic across the route. Traffic is a very important and unavoidable circumstance which can dampen the daily routine and its solutions need to be updated continually. Various reasons contribute to traffic. It is a wider categorisation. We have often come across bottlenecks which occur as a result of a wider road leading into a narrower one. This can lead to extension of the pile of vehicles and also occupy other lanes as well. The other reason for clogging can be improper lane changing. Considering a road with multiple civil transport vehicles and other private ones, these reasons just progressively add to traffic. Negligent driving sometimes lead to accidents which can just bring the entire stretch to a halt. Therefore, people should have an overview of the traffic intensity before they plan their journey. They should be able to decide between taking the public transport or their own vehicles. This leads to a requirement of a system which has the ability to capture, store, and analyse the traffic intensity. Any user can be prompted and advised on regular traffic updates which will help them decide, in advance or during transit, on their plans or alternate routes.

Neural Networks are preferable over deterministic programming for accurate and dynamic solutions and this traffic problem can be solved using this approach. Our proposed technique employs an offline process to predict the best source-to-destination route from a given set of parameters. It is a personalised application that has a wider usability and gets its data sets from Google Maps, which is one of the most trusted API for this purpose.

II. LITERATURE SURVEY

Congestion level estimation techniques for various type of data are our most related field. The study in [1] fed the data collected using a GPS device to a decision tree learning algorithm. The traffic congestion levels were defined by capturing the images of road traffic. In a similar study in [2], mobile phones are equipped with traffic apps that detect location using GPS and this information is sent to a remote server that predicts traffic congestion which is then passed on to the end user's phone. The study in [3] compares different supervised learning algorithms to identify transportation modes. The research demonstrates how to apply KNN, SVM, decision tree and RF wherein RF and SVM were found to produce best performances. In most studies the data is collected using GPS, accelerometer, gyroscope, mobile sensors. The studies in [4] and [5] predict lane change behaviour by collecting data from individual drivers' driving patterns and predicting if lane change initiation should be done or not. SVM classifiers were trained in [4] according to individual drivers' traits wherein the system produced error rate of 6%-7%. Various machine learning approaches like feed-forward neural network, recurrent neural network and support vector machines were compared in [5] and SVM resulted in best performance. A software library i.eTensorFlow is used in [6] where the model is trained by deep learning algorithm to predict traffic congestion. It used TPI to tell apart congested traffic conditions from non-congested traffic conditions. The system could estimate traffic congestion with 99% accuracy proving TensorFlow deep learning to be highly accurate. Similarly, the research in [7] predicts the road types and traffic congestions levels using neural network. The study in [8] detects the number plates of vehicles using machine learning techniques. A basic image processing techniques is used to extract possible objects from the number plates. Later, SVM classifiers, logistic regression and adaboost classifier were used trained to detect the number plate.

A few existing systems that try methods somewhat differingfrom the proposed system were compared to highlight differences.It shows how the proposed system can be considered tobe a better and more promising solution. Table I summarizesthem.

Existing System	Proposed System
Classification of Road traffic Congestion Levels from GPS data using a decision tree algo and sliding windows -- Captured [1] Images of road traffic by a video camera mounted on a test vehicle's dashboard to categorize traffic intensity --Utilized Sliding window protocol	--Traffic intensity classified by color densities on the traffic map screenshots. --No sliding window.
Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data Data was obtained [3] from smartphone sensors such as accelerometer, gyroscope, and rotation vector which were found to have important information for the purpose of mode recognition.	--Our system uses the Google API which already recognizes and provides us with the different modes. --For positioning, specifically GPS sensor is used to locate the user.
An automated signalized junction controller that learns strategies from a human expert	--Our system does not handle and alter the traffic system. It calculates the best

Consists of Hierarchy [9] of agents in the auctioning agent control system.	possible path from a source to a destination. --The junction parameter factors in the algorithm on the physical end of a user's device.
Real-time data visualization and machine learning for London traffic analysis. Google Cloud Platform [10] used for storage and processing of data from 14,000 traffic sensors across London.	--Our system does not deploy any sensors to gather data sets. Data extraction from the API and simply routing. A user may not travel at any given instant.

TABLE I. COMPARISON OF PROPOSED SYSTEM WITH EXISTING SYSTEMS

III. PROPOSED SYSTEM

The proposed system extracts the data set from screenshots obtained from Google Maps using Image Processing techniques. The extracted data contains features including location, time, and roadtype.

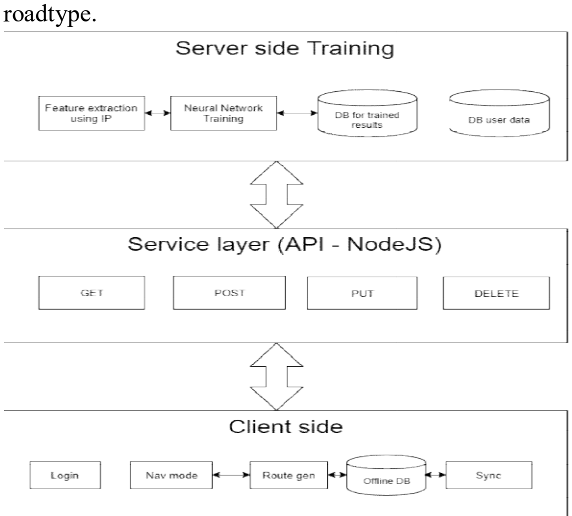


Fig. 1

Enough data is collected to have ample amount training samples with respect to each feature. The neural network is trained with this data set and the trained results get stored on a database present on the server. The trained results get sent to client applications where authorized users have logged in. The client application uses the trained network to predict traffic in a given area. To reduce traffic issues, the application provides alternate routes to a destination which have lesser traffic usually. To do this, the application evaluates routes segment wise using the neural network predictions. The route resulting with minimal predicted traffic is shown to the user. In this way, over time, the traffic patterns would change and get distributed more evenly. The mentality of naming a route "short" based on distance alone will change, allowing people to consider the probability of that route having traffic. Through regular updates of re-trained network, the client applications have the most up to date information about the city traffic patterns.

IV. SYSTEM ARCHITECTURE

The system architecture shown in Fig. 1 is split into threeparts; two major processing sites are the server side and clientside. A service layer bridges these two providing authorizationand access control on server data.

The server side consists of ML training processes and maintains the results on a database. The server talks with an external API, Google Maps API to obtain screenshots of chosen areas with real-time traffic being shown in them. Theclient side is an Android application giving a UI to the usertrying to get to a destination as quickly as possible with thehelp of the Neural Network algorithms traffic analysis. Bydefault the client application will display traffic conditions onthe map when the application is launched. This information iscalculated from the neural network as well.

V. CONCLUSION

Vehicular Traffic is an alarming issue that needs to be dealt with a solution quickly achievable. With the proposed machine learning technique, which will be available to all via an android application, the problem can be mitigated in a dynamic manner. The solution is kept personal to cater to individual immediate needs of going from one place to another. It does not put a big toll on the devices' performance as it works offline and avoids huge battery consumptions. Currently, this solution lends data from the Google Maps API; alternative sources of data can be tested to improve the system. The implementation is targeting only Android OS; in future, other prominent mobile OS can be taken into consideration to expand the solution's reach.

REFERENCES

[1] ThammasakThianniwet, SatidchokePhosaard and WasanPattara-Atikom, *Classification of Road Traffic*

Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows, Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009.

[2] Cynthia Jayapal, Sujith Roy. S, *ROAD TRAFFIC CONGESTION MANAGEMENT USING VANET*,International Conference on Advances in Human Machine Interaction (HMI - 2016), March 03-05, 2016.

[3] ArashJahangiri and Hesham A. Rakha, *Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data*, IEEE, 2015

[4] CharlottVallon, ZiyaErcan, Ashwin Carvalho and Francesco Borrelli, *A machine learning approach for personalized autonomous lane change initiation and control*, IEEE Intelligent Vehicles Symposium (IV), 2017

[5] UrunDogan and Johann Edelbrunner and IoannisIossifidis, *Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior*,Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011.

[6] HongsukYi, HeeJin Jung, Sanghoon Bae, *Deep Neural Networks for Traffic Flow Prediction*, IEEE, 2017.

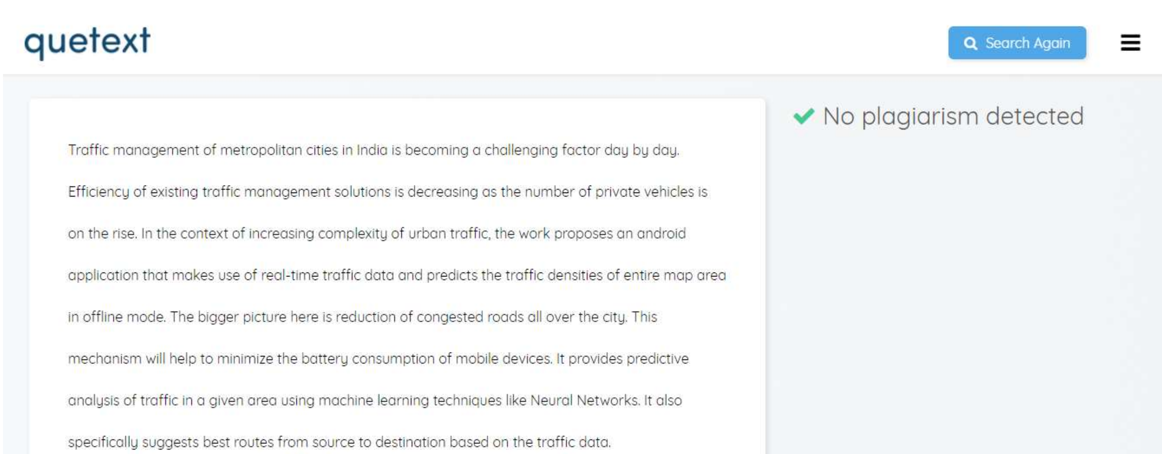
[7] Jungme Park, Zhihang Chen, Leonidas Kiliaris, Ming L. Kuang, M. AbulMasrur, Anthony M. Phillips, and Yi Lu Murphey, *Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 9, NOVEMBER 2009

[8] Anurag Sharma, Anurendra Kumar, K.V Sameer Raja, ShreeshLadha, *Automatic License Plate Detection*, IIT Kanpur, 2015-2016.

[9] Simon Box and Ben Waterson, *An automated signalized junction controller that learns strategies from a human expert*, UK, 2012

[10] Justin Kestelyn, *Real-time data visualization and machine learning for London traffic analysis*, November 2016

10.3.3 Plagiarism report of the paper

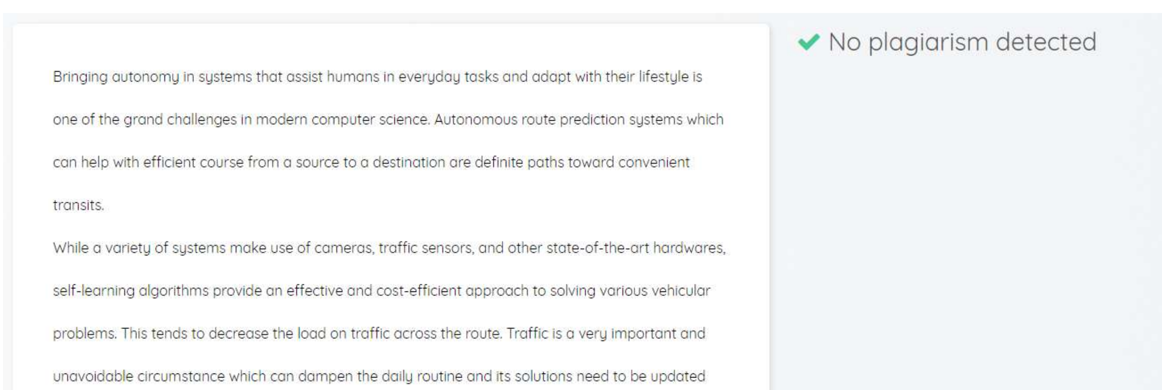


quetext

Search Again

No plagiarism detected

Traffic management of metropolitan cities in India is becoming a challenging factor day by day. Efficiency of existing traffic management solutions is decreasing as the number of private vehicles is on the rise. In the context of increasing complexity of urban traffic, the work proposes an android application that makes use of real-time traffic data and predicts the traffic densities of entire map area in offline mode. The bigger picture here is reduction of congested roads all over the city. This mechanism will help to minimize the battery consumption of mobile devices. It provides predictive analysis of traffic in a given area using machine learning techniques like Neural Networks. It also specifically suggests best routes from source to destination based on the traffic data.



No plagiarism detected

Bringing autonomy in systems that assist humans in everyday tasks and adapt with their lifestyle is one of the grand challenges in modern computer science. Autonomous route prediction systems which can help with efficient course from a source to a destination are definite paths toward convenient transits.

While a variety of systems make use of cameras, traffic sensors, and other state-of-the-art hardwares, self-learning algorithms provide an effective and cost-efficient approach to solving various vehicular problems. This tends to decrease the load on traffic across the route. Traffic is a very important and unavoidable circumstance which can dampen the daily routine and its solutions need to be updated quetext

Search Again

No plagiarism detected

Congestion level estimation techniques for various type of data are our most related field. The study in [1] fed the data collected using a GPS device to a decision tree learning algorithm. The traffic congestion levels were defined by capturing the images of road traffic. In a similar study in [2], mobile phones are equipped with traffic apps that detect location using GPS and this information is sent to a remote server that predicts traffic congestion which is then passed on to the end user's phone. The study in [3] compares different supervised learning algorithms to identify transportation modes. The research demonstrates how to apply KNN, SVM, decision tree and RF wherein RF and SVM were found to produce best performances. In most studies the data is collected using GPS, accelerometer,

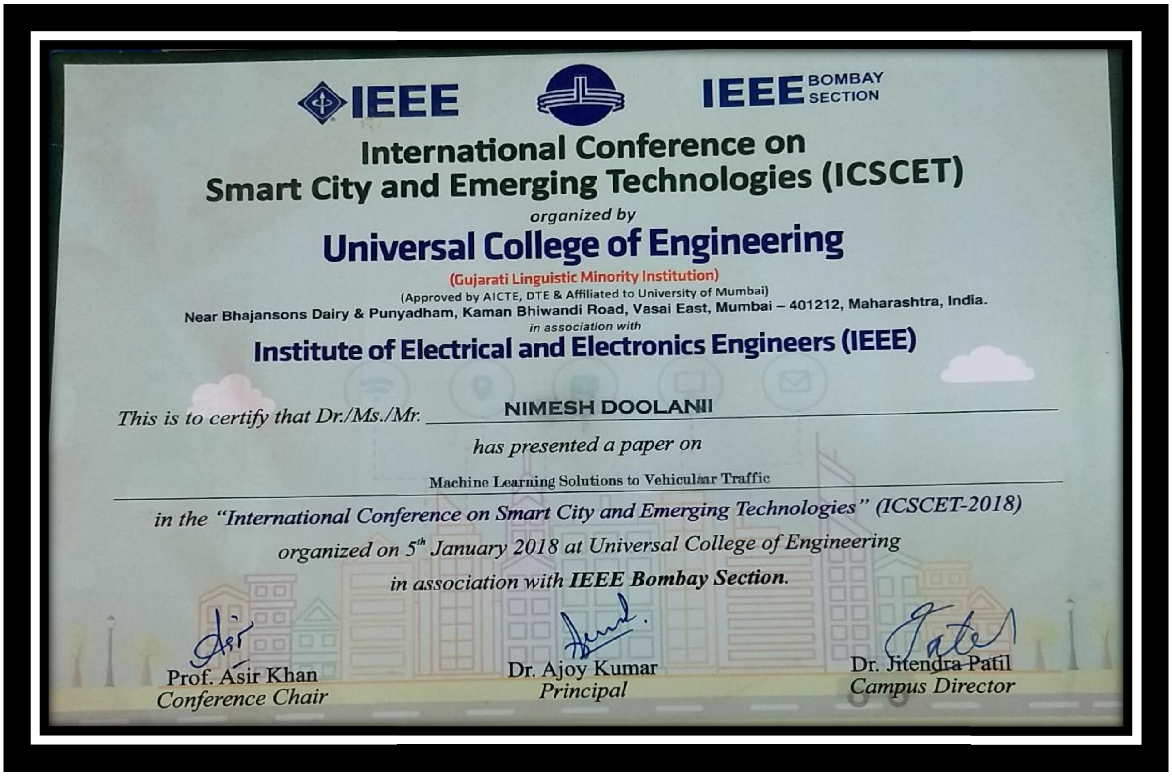
quetext

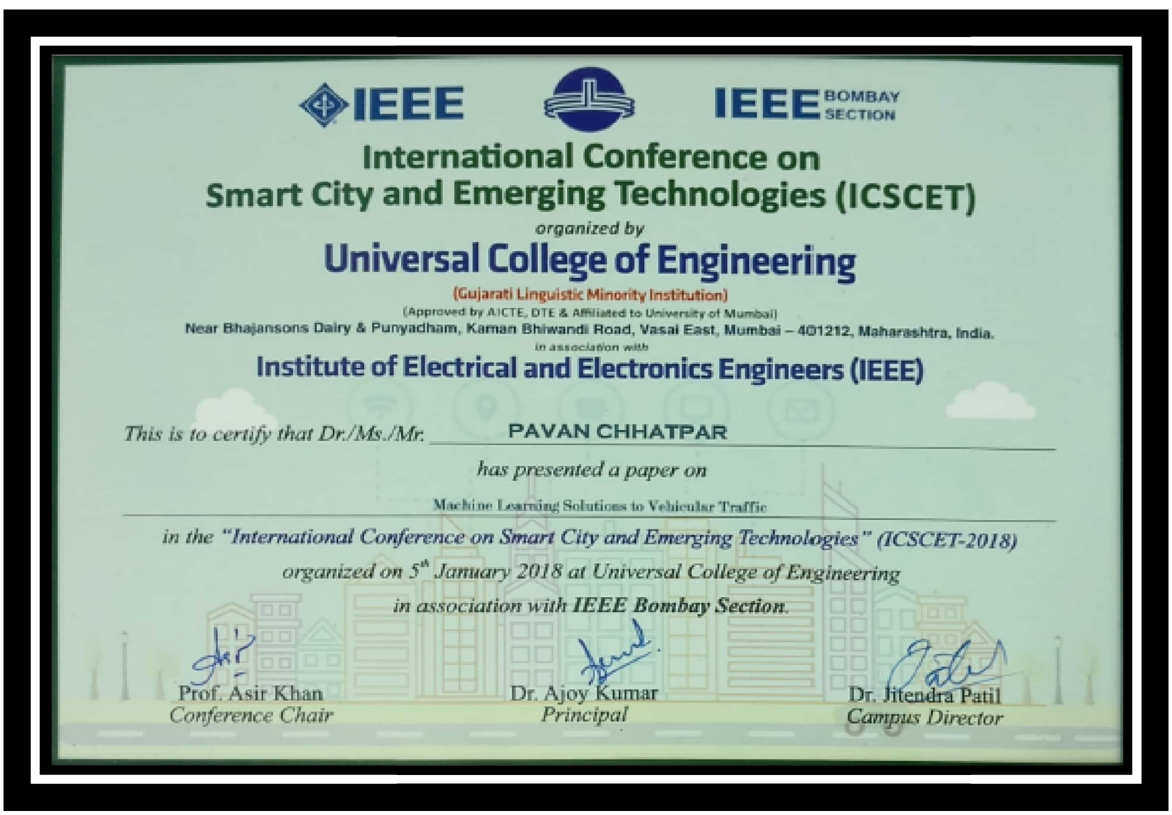
Search Again

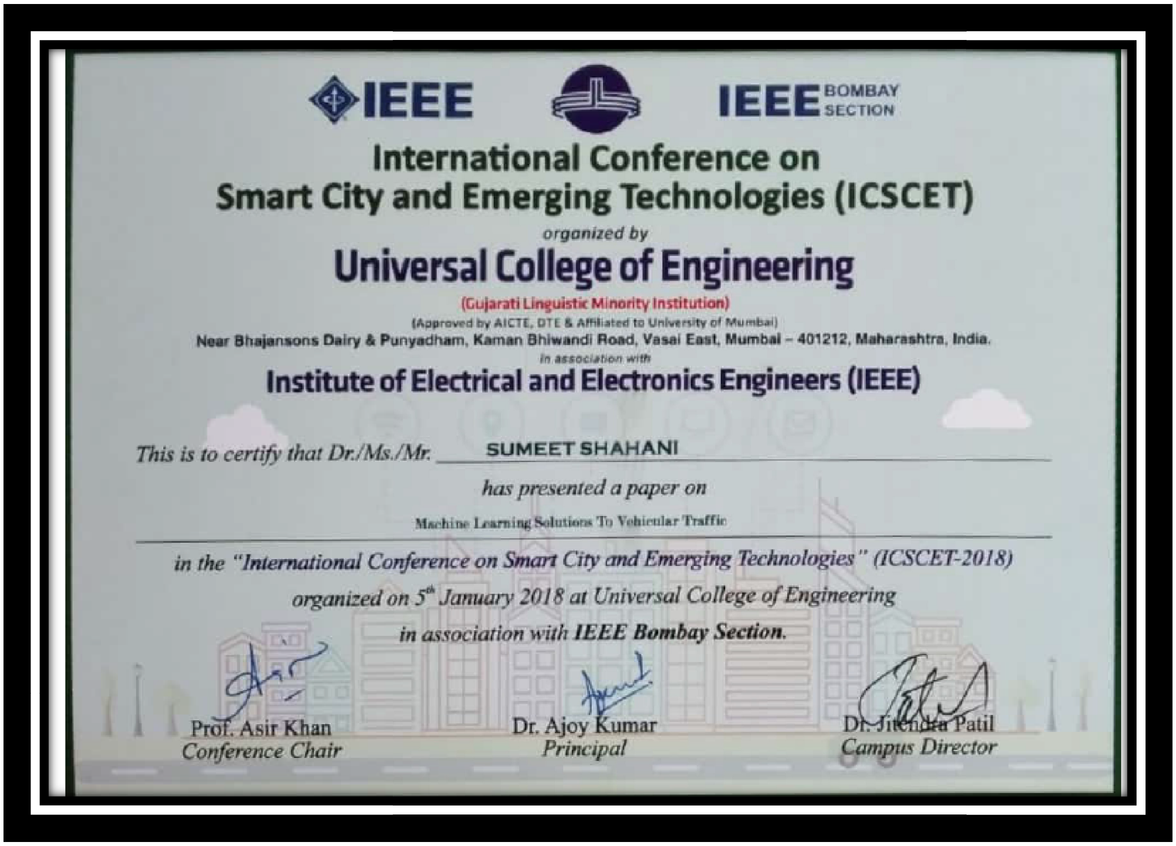
No plagiarism detected

The proposed system extracts the data set from screenshots obtained from Google Maps using Image Processing techniques. The extracted data contains features including location, time, and road type. Enough data is collected to have ample amount training samples with respect to each feature. The neural network is trained with this data set and the trained results get stored on a database present on the server. The trained results get sent to client applications where authorized users have logged in. The client application uses the trained network to predict traffic in a given area. To reduce traffic issues, the application provides alternate routes to a destination which have lesser traffic usually. To do this, the

10.3.4 Certificate of presentation of paper-I







10.3.5 Draft of the paper-II

Predictive Vehicular Traffic Model Using Ensemble Methods

Priya R.L, Pavan Chhatpar, Nimesh Doolani, Aysha Jagiasi,
and Sumeet Shahani, Computer Engineering, VESIT

Abstract-- **Traffic will always be a problem as we move to a modernised way of living. And in those stages of evolution, the complexity increases which in turn demands even more sophisticated measures to tackle them. Machine learning techniques have always helped in making efficient systems with minimal continual human intervention. A similar system is described in this paper. The data was collected from Google Maps. A processing technique was applied to extract features from the raw data. A significant amount of effort was put towards data preparation to make the features suitable for the proposed system. Multiple approaches to model the traffic data were tested before finalizing the model which best suited this proposal. For this research, a few selected areas were chosen. The system built is scalable and can be expanded to cover more areas.**

Keywords: **Machine Learning, Data Mining, Geohash, Traffic congestion, Route generator.**

I. Introduction

Traffic is a very important and unavoidable circumstance which can dampen the daily routine and its solutions need to be updated continually. Various reasons contribute to traffic. It is a wider categorisation. We have often come across bottlenecks which occur as a result of a wider road leading into a narrower one. This can lead to extension of the pile of vehicles and also occupy other lanes as well. The other reason for clogging can be improper lane changing. Considering a road with multiple civil transport vehicles and other private ones, these reasons just progressively add to traffic. At times, negligent driving leads to accidents which bring the entire stretch to a halt.

Therefore, people should have an overview of the traffic intensity before they plan their journey. They should be able to decide between taking the public transport or their own vehicles. This leads to a requirement of a system which has the ability to capture, store, and analyse the traffic intensity. Any user can be prompted and advised on regular traffic updates
Which will help them to decide, in advance or during transit, on their plans or alternate routes.

Bringing autonomy in systems that assist humans in everyday tasks and adapt with their lifestyle is one of the grand challenges in modern computer science. Autonomous route prediction systems which can help with efficient course from a source to a destination are definite paths toward convenient transits. While a variety of existing systems make use of cameras, traffic sensors, and other state-of-the-art hardwares, self-learning algorithms provide an effective and cost-efficient approach in solving various vehicular problems. This tends to decrease the load on traffic across the route.

But the main obstacle arises from the method of implementation of these systems. The tools applied are limited to abundant funding research facilities and therefore cannot be scaled to a cost-efficient and a larger geographical area with ever increasing traffic parameters.

Machine Learning techniques are more preferable over deterministic programming for accurate and dynamic solutions and this traffic problem can be solved using this approach. They have the ability to map and model into complex and nonlinear relationships. The proposed system demonstrates a process to predict the best source-to-destination route from a given set of parameters. It addresses a wide range of use cases and gets its data sets from Google Maps, which is one of the most trusted API for this purpose.

II. Literature Survey

Congestion level estimation techniques for various type of data are our most related field. The study in [1] fed the data collected using a GPS device to a decision tree learning algorithm. The traffic congestion levels were defined by capturing the images of road traffic. In a similar study in [2], mobile phones are equipped with traffic apps that detect location using GPS and this information is sent to a remote server that predicts traffic congestion which is then passed on to the end user's phone. The study in [3] compares different supervised learning algorithms to identify transportation modes. The research demonstrates how to apply KNN, SVM, decision tree and RF wherein RF and SVM were found to produce best performances. In most studies the data is collected using GPS, accelerometer, gyroscope, mobile sensors. The studies in [4] and [5] predict lane change behaviour by collecting data from individual drivers' driving patterns and predicting if lane change initiation should be done or not. SVM classifiers were trained in [4] according to individual drivers' traits wherein the system produced error rate of 6%-7%.

Various machine learning approaches like feed-forward neural network, recurrent neural network and support vector machines were compared in [5] and SVM resulted in best performance. A software library i.eTensorFlow is used in [6] where the model is trained by deep learning algorithm to predict traffic congestion. It used TPI to tell apart congested traffic conditions from non-congested traffic conditions. The system could estimate traffic congestion with 99% accuracy proving TensorFlow deep learning to be highly accurate. Similarly, the research in [7] predicts the road types and traffic congestions levels using neural network.

The study in [8] detects the number plates of vehicles using machine learning techniques. A basic image processing techniques is used to extract possible objects from the number plates. Later, SVM classifiers, logistic regression and adaboost classifier were used trained to detect the number plate. The system in [9] used hardware devices throughout the city to detect traffic, whereas our proposed system uses data from Google Maps which use crowdsourcing to collect traffic information. Crowdsourcing does not require dedicated hardware devices to be installed, they depend on mobile phones, which nearly everyone owns today.

The work in [10] makes use of GPS + Wifi data from vehicles to evaluate the traffic conditions. The agents that are monitoring the traffic calculate a bid based on the position and speed of the vehicles. Based on this bid, the traffic light on the roads is changed. The neural network can either be trained by human input or reinforcement learning by temporal difference.

In [11], a variety of sensors were deployed for tracking the vehicular movements on highways while on arterial networks, sparsely sampled and high frequency GPS techniques were used. Various fleet delivery vehicles and volunteers helped in sharing valuable transit data.

III. PROPOSED SYSTEM

The proposed system's architecture is shown in Fig. 6. It begins with the data collection in the form of images on which the processing has to be applied. The entire collection is passed through the Image Processing phase which extracts all crucial information required to perform analysis. The extracted data goes through transformation and augmentation of features. Further, ExtraTree Classifier trains over the 80% of the dataset and eight trained models, where each is a collection of ten decision trees, are saved on the server in JSON format which can be downloaded by authorized clients. The client application uses the trained models to predict traffic in a given area. As an example use case, the application provides alternate routes to a destination which have lesser traffic as per the prediction. Many more use cases are present like, fleet management can observe traffic of upcoming days to schedule fleets accordingly, in general any scenario involving travel can be planned ahead of time by taking into account the traffic possible in future. For the current use case being implemented, the application evaluates routes segment-wise using the predicted traffic. Locations for the segments are readily available and get converted to Geohash values. Time is taken as the current time for the first segment and it gets incremented as the algorithm iterates through the remaining segments. The resultant intensity between two points can be calculated as considering the traffic mode from the initial point till the midpoint as one segment and the mode of the second point considered from the midpoint till that point. Distance and hence the midpoint can be calculated by the haversine formula. The four traffic classifications are assigned with some speeds which differ on different lane roads. The distance and the speed pairs between the two points can help us arrive to an average time required to travel from one point to other. The estimated time of arrival(ETA) on a route can be calculated by summing all the average time calculated between pairs. This is again looped through all the alternate routes and the one with the lowest ETA is shown highlighted as the best route. The mentality of naming a route "short" based on distance alone is corrected by adding traffic as an influencing factor.

The other aspect to be taken into consideration was the limited area that was taken into account for processing and learning. A source or destination could be easily out of the area and the algorithm would not give sound inference. This is avoided by geofencing the supported areas. Any location found outside the considered region would simply trigger a notification to the user prompting them that location is out of service. This also features into the routing algorithm. Any route that took the transit from outside the region would simply be discarded.

Once this sample model was optimised and utilised, the area for implementation could be increased to a vast region with possibly more classification of complex roads. Still, the concept of geofencing helps provide accuracy and legitimacy into the system when a large region cannot be incorporated into the model.

Through regular updates of re-trained model, the client applications have the most up to date information about the city traffic patterns.

A. Data Collection Methodology

Data was collected over a month at every fifteen minutes for two selected regions in Mumbai namely, Chembur and Ghatkopar. This data is in pictorial form and needs to be processed to extract features from it. Further the processed data has to be cleaned and transformed to a suitable format before testing it against Machine Learning algorithms. Each tested approach consists of Image Processing, data cleaning and transformation, and finally a Machine Learning algorithm to learn from the prepared data. A total of two approaches are defined uptil the stage of preprocessing. Further multiple Machine Learning algorithms are evaluated for its performance on the data.

B. System Block Diagram

The entire process of system is shown in Fig. 1 as a Block Diagram. In the Block Diagram, there is one external interface, Google Maps API, and four modules, Screenshot generator, Image Processing, Geohash formation and ExtraTree Classifier training which works on server side. The results of these are stored on a database will be later used by client application. Route generator takes user input through the application and gives the best route using the trained models' predictions. It takes source and destination addresses, records journey time and splits up alternative routes into segments for calculating predictions.

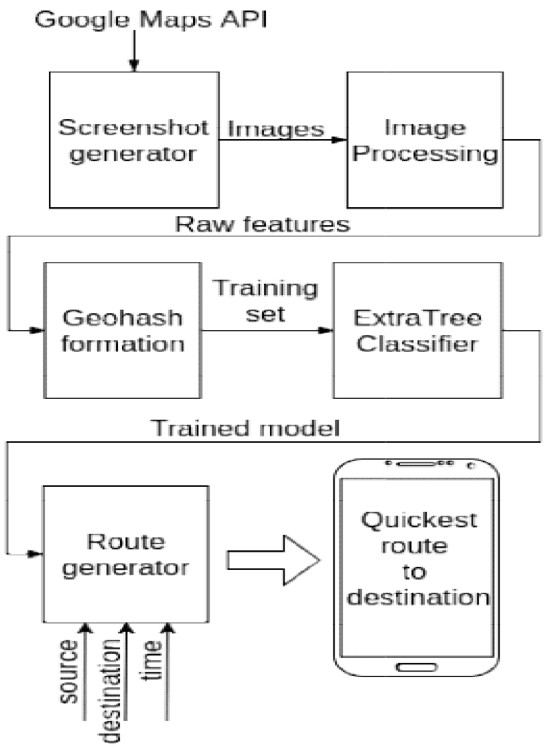


Fig. 1 Block Diagram

IV. IMPLEMENTATION DETAILS

The implementation of the system is composed of the following brief stages. In each stage alternative methodologies have been tried to choose the best one.

A. Preprocessing

The first approach uses an unbiased approach for feature extraction. The pictures are filtered to remove all colors except those representing traffic. Next, sampling traffic begins. The features noted are the latitude, longitude obtained via shift of origin and scaling applied over pixel coordinates of the image. Other than these, time is noted as weekday, hour and minute obtained from creation time of the picture. The target value is one of the four classes, light traffic, medium traffic, heavy traffic and very heavy traffic, based on the corresponding color shown on that coordinate. Without any further constraint, a huge amount of tuples might get extracted from a single picture. To avoid this situation, an upcoming latitude, longitude pair is considered only if there is no other existing sample in a 50m radius with the same target traffic. The minute's value for each recorded tuple was floored down to the nearest 15th minute; this was necessary because the interval while capturing pictorial data was not exactly 15 minutes due to inherent system delay in triggering actions and fluctuating internet speeds required to load the map and traffic.

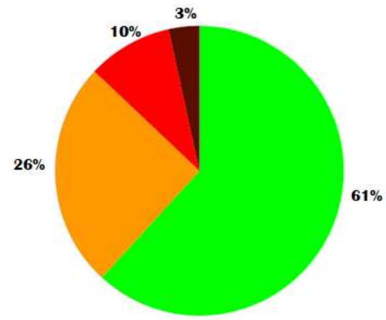


Fig. 2 Initial class distribution for first approach

The problem with the above technique is the highly imbalanced distribution of target traffic in the tuples extracted as seen in the Fig. 2. The major class covers roughly 60% of the data while the minor class covers only 3% of data. To tackle this, the minor class was discarded and two major classes were reduced to the size of the third class. This reduction was achieved by clustering the data of each of the two major classes into the same number of clusters as the size of the third class. Clustering was achieved by K-Means which is an unsupervised Machine Learning algorithm with an ability to cluster data into a said number of clusters. This process is time consuming as the total number of tuples are nearly 4,000,000 and two sets of 400,000 clusters are to be prepared from a set of 2,500,000 and 1,000,000 tuples. The resulting balanced classes are seen in Fig. 3.

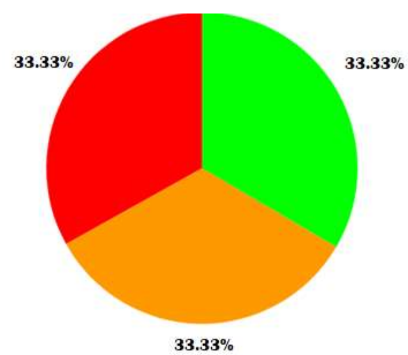


Fig. 3 Balanced class distribution for first approach

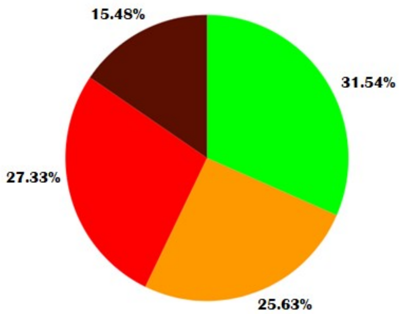


Fig. 4 Class distribution in second approach

In the second approach, the problem of imbalance is addressed at the ground stage itself by using a biased method to extract features from images. The entire Image Processing remains same, except that instead of sampling all target classes' tuples at a fixed 50m, the major class is sampled at larger distances and minor class is sampled at very small distances apart. The gives us a dataset which is inherently less imbalanced and does not require clustering. All four target classes are usable and have good number of samples each as seen in Fig. 4. The imbalance issue doesn't get completely solved here; during training in the further steps, the algorithm will be informed about weights of each class, dependent on ratio of the class in the dataset, which will be multiplied with the loss function to penalize the minor class more than the major one. This basically forces the algorithm to learn the minor class with more effort.

There are approximately 4,000,000 tuples of data and with just 5 features in hand the model might end up underfitting. The latitude and longitude values form a very complex distribution which is tough to map by any math function; it will also not be scalable if new regions are used to train the algorithm. So, both these features are converted into a Geohash[12], which is a string of length 7. The Geohash algorithm assigns a hash value to all geolocations in the world and its precision is adjustable. The maximum precision available is +-11cm at a string length of 12. A string of length 7 gives precision of +-70m. The remaining features of weekday, hour and minutes are augmented using their squares as added features. So a total of 7 features are now available for the dataset.

The traffic patterns change vastly with time and applying same parameters to the entire 24 hour cycle might not give best results. So periods of 3 hours are taken generating 8 datasets over a period of 24 hours. Each dataset will be learnt by an individual model.

B. Machine Learning Algorithms

After preprocessing with the first approach which was seemingly unconventional due to its high time consumption and possible loss of valuable information due to K-Means clustering; just one algorithm was tried over the data. A deep neural network was used to train over the dataset. The data was normalized in the range of [-1,1] which is known to be well suited for neural networks. The model had 3 hidden layers with 10 neurons each, using the ReLU activation function. The learning rate was set to 0.1. Training happened using 100 complete epochs over mini batches of 1/8th of the total dataset, giving a total of 800 iterations. A major influencing factor in a neural network is the optimizing algorithm; the simple Gradient Descent algorithm failed to converge. The Adagrad (Adaptive Subgradient) algorithm[13], allows different step sizes, while moving towards to the optima, for different features. This facilitates a good amount of influence of rare but informative features. It varies step sizes by adjusting the learning rate as it moves closer to an optima. This algorithm was used for optimization of the loss function for our model.

All of the following algorithms were trained using the dataset created by the second approach of preprocessing. Each algorithm generated 8 models as the dataset had been split into subsets covering 3 hours each. As mentioned earlier, each algorithm will penalize the loss function as per the weight of the target class for the tuple currently under consideration. The first attempt made was to use SVM (Support Vector Machine) with linear, poly and rbf kernels. The linear kernel performed well in the learning stage. The loss function used here was the squared hinge loss, which is literally the square of the standard hinge loss function. SVM provides a parameter C, which is the default penalty parameter for the loss function. It helps to control overfitting of the algorithm. A value of 2.5 was used here along with the class weights together.

The dataset present, could be considered of having all categorical features. Geohash was inherently categorical; the remaining features of weekday, hours and minutes along with their squares had a finite set of possible values, 7, 24 and 4 respectively. Hence, Bernoulli Naive Bayes with an additive smoothing parameter of 0.5 was given a run too. Amongst the ensemble methods of Decision Trees, Random Forest was used. Decision Trees overfit immensely in unconstrained environments, hence the tree depths were limited to 20 and the forest had 10 estimators (trees).

There is one overfitting problem with Random Forest which exists due to the way its algorithm works. It uses a number of trees to learn sub samples of data which are decided from bootstrap tuples fixed at the beginning of the algorithm. To overcome this, Extra Trees algorithm[14] was tried which chooses samples from the entire training set for deciding the variables at the split as opposed to Random Forest which uses only the bootstrap samples. Other parameters of the algorithm were kept unchanged to avoid excessive deepening of the trees.

V. RESULTS

Having discussed the available alternatives, performance of each, post the training, was evaluated on an unseen test set. The five different algorithms used are henceforth referred as K-Means-NeuralNet, Geohash-LinearSVM, Geohash-BernoulliNB, Geohash-ExtraTrees and Geohash-RandomForest.

The accuracies were calculated as the ratio of correct predictions with total predictions. In terms of accuracy, as seen in Fig. 5 the winner was Geohash-ExtraTrees closely followed by Geohash-RandomForest. Decision trees were found to well suit the problem and as expected, ExtraTree algorithm generalized better than RandomForest.

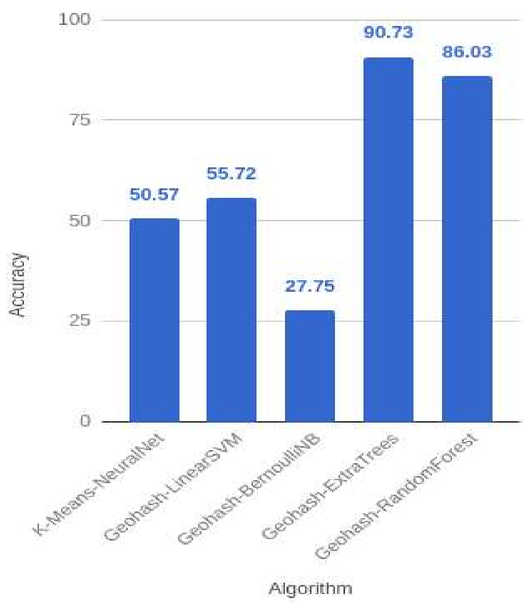


Fig. 5Comparing accuracies of alternative algorithms

Algorithm	Green	Orange	Red	Dark red
K-Means-NeuralNet	-5.32	-18.73	24.05	0
Geohash-LinearSVM	20.18	-16.75	-4.52	1.09
Geohash-BernoulliNB	- 24.64	-19.94	-7.31	51.89
Geohash-ExtraTrees	-0.21	-0.05	0.03	0.23
Geohash-RandomForest	1.43	-0.66	-0.55	-0.22

TABLE 1 Difference in expected and predicted distribution of output classes

Another evaluation of the algorithms was done by comparing the distribution of target classes in the test set with that of the predicted results. A general observation would be to get almost the same distribution in both cases; so the difference between ratio of actual presence of a class and ratio of predicted presence of that class should be zero or as close to zero as possible.

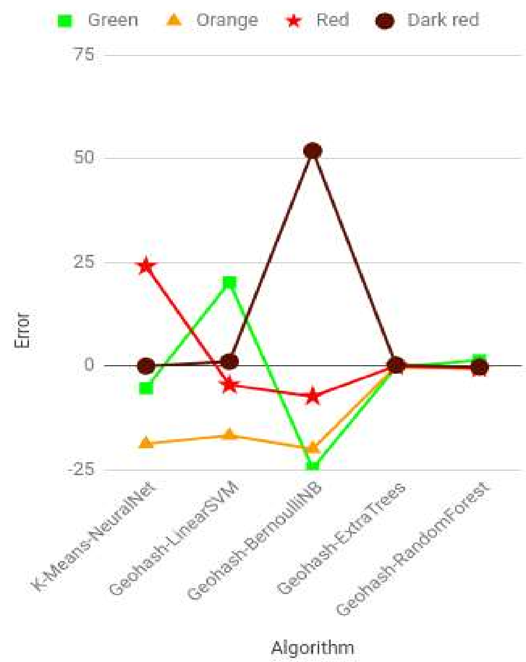


Fig. 6 Graphical view of difference in expected and predicted distribution of output classes

The results were collected from the confusion matrix prepared over the test set targets and predictions made over the test set features. Ratio of presence of each class was calculated as a percentage value. Looking at Fig. 6 generated from TABLE 1 it's seen that again, ExtraTree and RandomForest algorithms have worked the best.

VI. CONCLUSION

Vehicular Traffic is an alarming issue that needs to be dealt with a solution quickly achievable. With the proposed supervised learning technique, which will be available to all via an android application, the problem can be mitigated in a dynamic manner. The solution is kept personal to cater to individual immediate needs of going from one place to another. It does not put a big toll on the devices' performance as it works offline and avoids huge battery consumptions. It also helps to reduce the accident rate in the city.

Currently, this solution lends data from the Google Maps API; alternative sources of data can be tested to improve the system. The implementation is targeting only Android OS; in future, other prominent mobile OS can be taken into consideration to expand the solution's reach.

REFERENCES

[1] ThammasakThianniwet, SatidchokePhosaard and WasanPattara-Atikom, *Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows*, Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009.

[2] Cynthia Jayapal, Sujith Roy. S, *ROAD TRAFFIC CONGESTION MANAGEMENT USING VANET*, International Conference on Advances in Human Machine Interaction (HMI - 2016), March 03-05, 2016.

[3] ArashJahangiri and Hesham A. Rakha, *Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data*, IEEE, 2015

[4] CharlottVallon, ZiyaErcan, Ashwin Carvalho and Francesco Borrelli, *A machine learning approach for personalized autonomous lane change initiation and control*, IEEE Intelligent Vehicles Symposium (IV), 2017

[5] UrunDogan and Johann Edelbrunner and IoannisIossifidis, *Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior*, Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011.

[6] Hongsuk Yi, HeeJin Jung, Sanghoon Bae, *Deep Neural Networks for Traffic Flow Prediction*, IEEE, 2017.

[7] Jungme Park, Zhihang Chen, Leonidas Kiliaris, Ming L. Kuang, M. AbulMasrur, Anthony M. Phillips, and Yi Lu Murphey, *Intelligent Vehicle Power Control Based on Machine Learning of Optimal Control Parameters and Prediction of Road Type and Traffic Congestion*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 9, NOVEMBER 2009

[8] Anurag Sharma, Anurendra Kumar, K.V Sameer Raja, ShreeshLadha, *Automatic License Plate Detection*, IIT Kanpur, 2015-2016.

[9] Justin Kestelyn, *Real-time data visualization and machine learning for London traffic analysis*, November 2016

[10] Simon Box and Ben Waterson, *An automated signalized junction controller that learns strategies from a human expert,* Draft of the paper in Engineering Applications of Artificial Intelligence, 2012

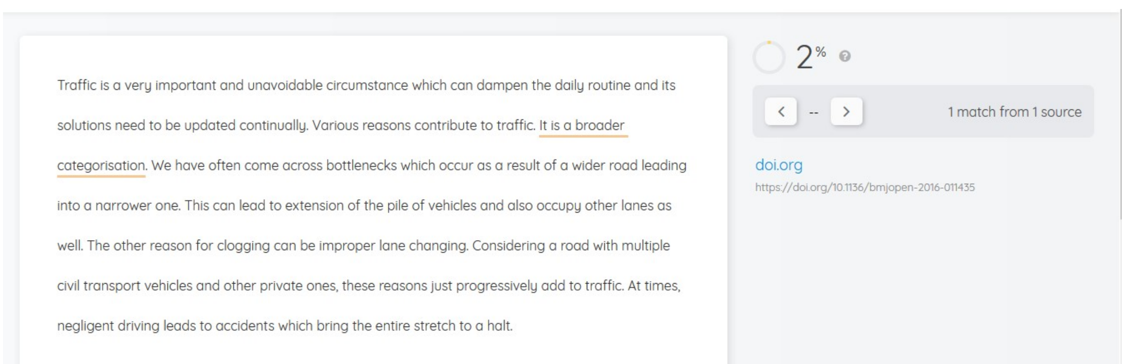
[11] Ryan Jay Herring, "Real-Time TracModeling and Estimation with Streaming Probe Data using Machine Learning", Fall 2010

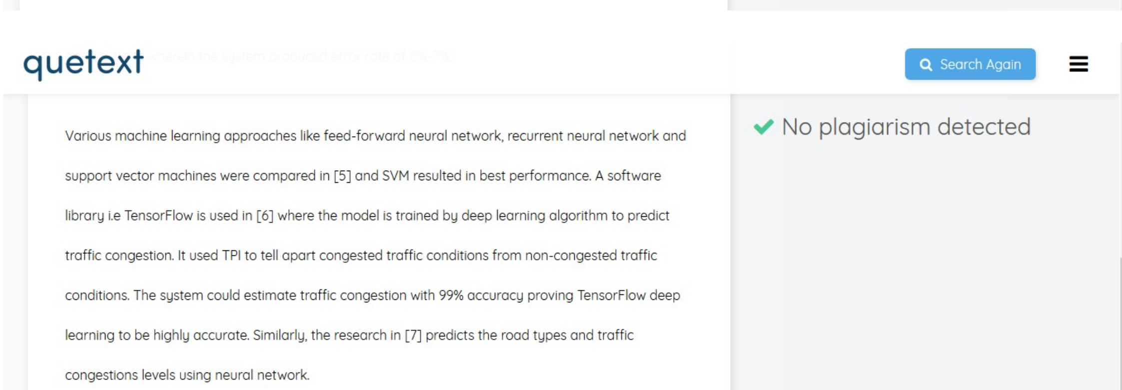
[12] Geohash.org. (2018). *Geohash - geohash.org*. [online] Available at: http://geohash.org/

[13] Duchi, John, EladHazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of Machine Learning Research* 12.Jul (2011): 2121-2159.

[14] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." *Machine learning* 63.1 (2006): 3-42.

10.3.6 Plagiarism report of paper-II





quetext

Search Again

✔ No plagiarism detected

Data was collected over a month at every fifteen minutes for two selected regions in Mumbai namely, Chembur and Ghatkopar. This data is in pictorial form and needs to be processed to extract features from it. Further the processed data has to be cleaned and transformed to a suitable format before testing it against Machine Learning algorithms. Each tested approach consists of Image Processing, data cleaning and transformation, and finally a Machine Learning algorithm to learn from the prepared data. A total of two approaches are defined uptil the stage of preprocessing. Further multiple Machine Learning algorithms are evaluated for its performance on the data.

A. Preprocessing

quetext

Search Again

✔ No plagiarism detected

After preprocessing with the first approach which was seemingly unconventional due to its high time consumption and possible loss of valuable information due to K-Means clustering; just one algorithm was tried over the data. A deep neural network was used to train over the dataset. The data was normalized in the range of [-1,1] which is known to be well suited for neural networks. The model had 3 hidden layers with 10 neurons each, using the ReLU activation function. The learning rate was set to 0.1. Training happened using 100 complete epochs over mini batches of 1/8th of the total dataset, giving a total of 800 iterations. A major influencing factor in a neural network is the optimizing algorithm; the simple Gradient Descent algorithm failed to converge. The Adagrad (Adaptive Subgradient) algorithm[13], allows different step sizes, while moving towards to the optima, for different features. This facilitates a good amount of influence of rare but informative features. It varies step sizes by adjusting the learning rate as it moves closer to an optima. This algorithm was used for optimization of

quetext

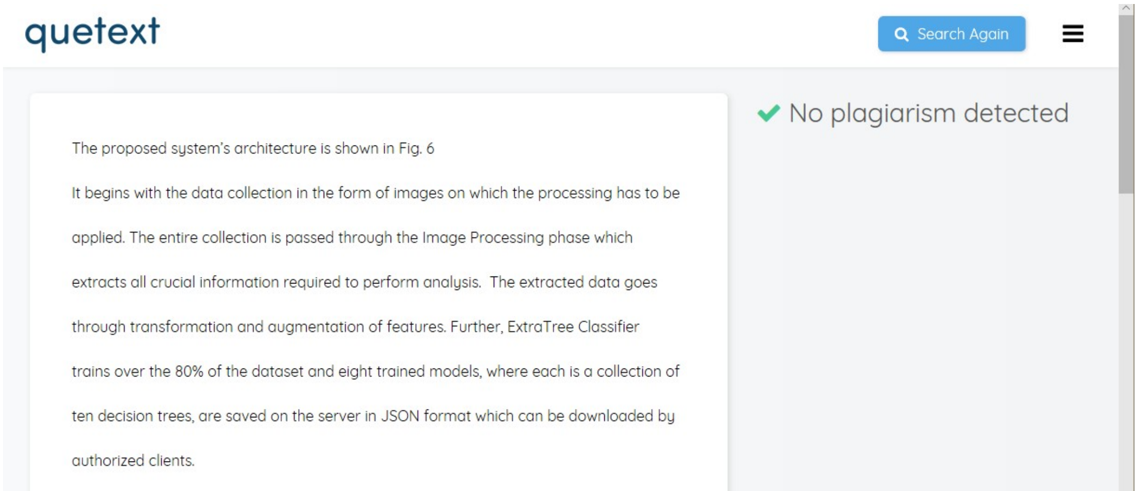
Search Again

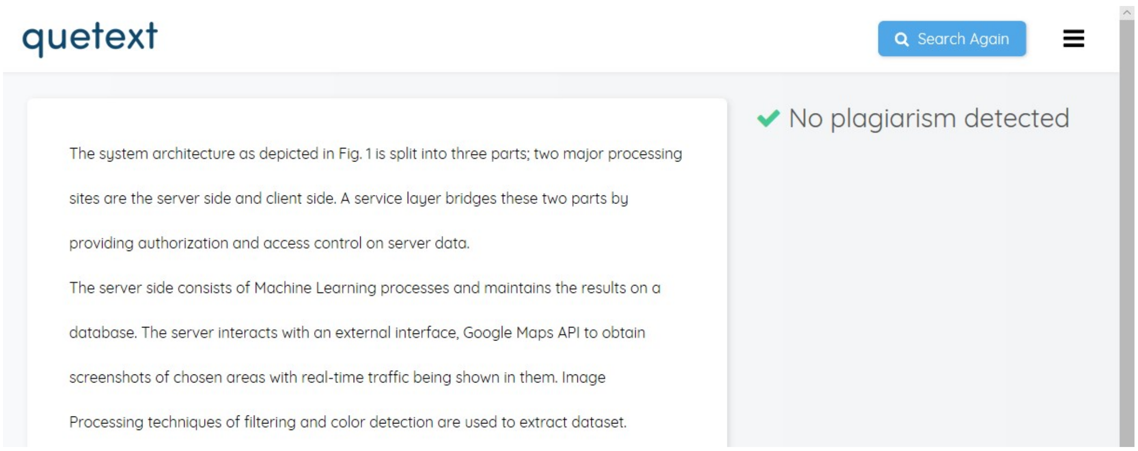
✔ No plagiarism detected

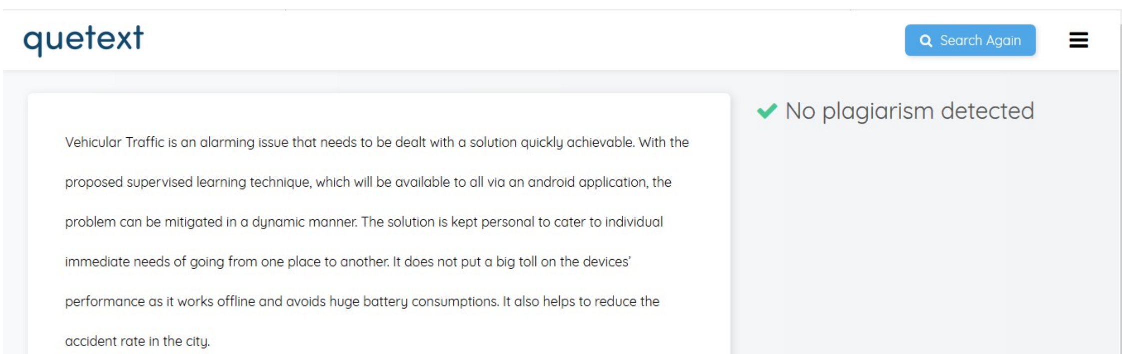
Having discussed the available alternatives, performance of each, post the training, was evaluated on an unseen test set. The five different algorithms used are henceforth referred as K-Means-NeuralNet, Geohash-LinearSVM, Geohash-BernoulliNB, Geohash-ExtraTrees and Geohash-RandomForest.

The accuracies were calculated as the ratio of correct predictions with total predictions. In terms of accuracy, as seen in Fig. 4 the winner was Geohash-ExtraTrees closely followed by Geohash-RandomForest. Decision trees were found to well suit the problem and as expected, ExtraTree algorithm generalized better than RandomForest.

Another evaluation of the algorithms was done by comparing the distribution of target classes in the test set with that of the predicted results. A general observation would be to get almost the same distribution in both cases; so the difference between ratio of actual presence of a class and ratio of


quetext
Search Again
No plagiarism detected
The proposed system's architecture is shown in Fig. 6
It begins with the data collection in the form of images on which the processing has to be applied. The entire collection is passed through the Image Processing phase which extracts all crucial information required to perform analysis. The extracted data goes through transformation and augmentation of features. Further, ExtraTree Classifier trains over the 80% of the dataset and eight trained models, where each is a collection of ten decision trees, are saved on the server in JSON format which can be downloaded by authorized clients.


quetext
Search Again
No plagiarism detected
The system architecture as depicted in Fig. 1 is split into three parts; two major processing sites are the server side and client side. A service layer bridges these two parts by providing authorization and access control on server data.
The server side consists of Machine Learning processes and maintains the results on a database. The server interacts with an external interface, Google Maps API to obtain screenshots of chosen areas with real-time traffic being shown in them. Image Processing techniques of filtering and color detection are used to extract dataset.


quetext
Search Again
No plagiarism detected
Vehicular Traffic is an alarming issue that needs to be dealt with a solution quickly achievable. With the proposed supervised learning technique, which will be available to all via an android application, the problem can be mitigated in a dynamic manner. The solution is kept personal to cater to individual immediate needs of going from one place to another. It does not put a big toll on the devices' performance as it works offline and avoids huge battery consumptions. It also helps to reduce the accident rate in the city.

10.3.7 Xerox of Project Review Sheet

Inhouse/ Industry: ✓

Class: D17 A/B/C ✓

Group No.: 1

Project Evaluation Sheet 2017 - 18

Title of Project: Vehicular Traffic Abatement

Group Members: Pavan Chhatpar (12), Nimesh Doolani (16), Aysha Jagiasi (23), Sumeet Shahani (64)

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environ ment Friendly	Ethics	Team work	Presentati on Skills	Applied Engg &Mgmt principles	Life - long learning	Profess ional Skills	Innov ative Appr oach	Total Marks
	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(5)	(5)	(50)
Review of Project Stage 1	5	5	5	3	4	2	2	2	1	3	2	3	4	4	45

Comments: A bit of work needed on presentation

Richard Joseph

Name & Signature Reviewer1

	Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environ ment Friendly	Ethics	Team work	Presentati on Skills	Applied Engg &Mgmt principles	Life - long learning	Profess ional Skills	Innov ative Appr oach	Total Marks
	(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(5)	(5)	(50)
Review of Project Stage 1	5	5	5	04	04	02	02	02	02	03	02	03	04	04	47

Comments: Add your process with i/p & its corresponding o/p for particular location in ppt

Date: 15th March,2018

Swati Sharma

Name & Signature Reviewer2

15/03/18