KNN Algorihtm Based Traffic Analysis And Prediction

Desai Vishwas, A.S Rakesh, Shetty Manish, Agarwal Sakshi

Department of Information Science and Engineering
BMS Institute of Technology and Management
Bangalore, Karnataka, India

***Abstract* - Road networks form backbones of any country's development structure. The free flow of road traffic is important for quicker connectivity and transport system. The sensors will be designed for analysing traffic and transport flow in a smart city lane. For various groups of road users, such as commuters, private vehicle travellers and the public transport system, accurate information on traffic flows is desperately needed. This knowledge would help road users make better choices in terms of travel, increase traffic quality , reduce emissions, and resolve traffic congestion. The aim of forecasting traffic congestion is to provide information about traffic congestion well in advance. The prediction of traffic congestion has gained popularity as Smart Transport Systems rapidly expanded and deployed**

In this paper we review prediction system for traffic congestion using IR sensors, embedded systems using Arduino, Blynk IOT, and an Android Application.

***Keywords*— Prediction, Traffic Congestion, KNN, Arduino, machine Learning, Blynk, Internet of Things, Android, Infra-red sensors, Cloud computing,**

I. INTRODUCTION

Road networks form backbones of any country's development structure. The free flow of road traffic is important for quicker connectivity and transport system. The sensors will be designed for analysing traffic and transport flow in a smart city lane. For various groups of road users, such as commuters, private vehicle travellers and the public transport system, accurate information on traffic flows is desperately needed. This knowledge would help road users make better choices in terms of travel, increase traffic quality, reduce emissions, and resolve traffic congestion. The aim of forecasting traffic congestion is to provide information about traffic congestion well in advance. The prediction of traffic congestion has gained popularity as Smart Transport Systems rapidly expanded and deployed. This prediction system for traffic congestion uses IR sensors, embedded systems using Arduino, a Wi-Fi module, and an Android Application Interface to predict and notify congestion along a specific path. Our Android App displays the congested route, forecasts the congestion level and severity, and notifies users of the congestion. It helps users who have registered for the app receive real-time notifications of any congestion that occurs in any part of town.enhance job execution, monitor health, and make available real-time statistics. In a smart city, the road would be equipped with a traffic flow analysis sensor. Therefore, free flow of road traffic is necessary for faster connectivity and transport systems. Traffic congestion affects many cities around the world, causing a number of problems, such as fuel waste, increased levels of stress, delayed deliveries, and monetary losses. As a result, accurate prediction of traffic jams is urgently needed to minimize those losses. Yet forecasting is a major challenge to produce promising results in urban networks for vibrant and unpredictable traffic flows. This research project proposes a new model of traffic jam prediction based on previous traffic results. Classification of online traffic continues to be of long-term interest to the networking community. This acts as feedback for realistic approaches such as network control, quality of service and intrusion detection. In this project, we present a machine-learning approach that uses a decision tree to precisely classify internet traffic. Accuracy is not our only concern; it is also extremely critical in terms of latency and throughput. Congestion is both burdensome and agonizing India has one of the largest road networks amongst all countries. The national highways cover almost 97,991 km of a total length of 5.4 million km of the road network. The main cause of traffic congestion is the large number of vehicles caused by growth and economic development. In the event of congestion, average urban residents spend more than 10 hours a week driving. Sensors will be designed to measure traffic flow on smart city roads, and there are still few traffic prediction approaches using neural networks and other prediction models that are not as effective for many real-world applications. Thus, this paper proposes a solution for traffic analysis using KNN algorithm that would select only part of the data to be analyzed as two-thirds of all data and predict traffic congestion on a specific path, and would notify vehicles intending to move along that specific path well in advance. Precise information on traffic flows therefore allow road users to move quickly and safely.

PROBLEM STATEMENT

Circumstance aware planning of routes gathers expanding enthusiasm as urban communities become swarmed and stuck. We present a framework for singular outing plans that consolidates future traffic perils in routing. Further, we make an application that gives the end client an expansive perspective on the traffic analysis that is being processed by our framework..

MOTIVATION

With this paper, we propose a model that can be utilized for examination and forecast of congestion by the frameworks installed at the intersection of lanes. We expect all the smart cities are very much evolved and well connected, with all the sensors at the critical intersections being deployed. The information is assembled from various sensors from different intersection points. The gathered information is accepted to be time-sensitive stream data. Our point is to predict the congestion on a certain lane that is going to happen in due time.

EXISTING SYSTEM

The existing system is where the traffic policemen must be present there in the flesh to resolve congestion. There is no proper way to resolve congestion in a city like Bengaluru. There are not enough human resources to manage and resolve the traffic congestion in Bengaluru, which is notoriously famous for its slow-moving traffic. But with changing times and technology, few prediction and congestion mitigation systems have been developed. Most of these systems do not work for a complex traffic system like Bengaluru. Most of these systems rely on the type of vehicle plying on the roads.

PROPOSED SYSTEM

In this project we propose a novel architecture which can be used by the systems deployed at the road junction for analysis and congestion prediction. We assume that all the smart cities are well developed and well connected, with all the sensors deployed at the crucial junctions. The information is being accumulated from various intersection points through various sensors. The information is thought to be a stream of real-time information that is time reliant. We will predict the congestion in a particular way which is going to happen in the due time. This framework predicts the blockage as well as offers a piece of the answer to relieving the jam.

The proposed system will be:

- Able to manage real-time information easily
- Faster usage with effective programming
- Easy to utilize interface for a wide range of clients
- A single dashboard with pertinent information for productive congestion strategies
- More confined to suit the requirements of the end client
- Robust

PAPER OBJECTIVE

The handling of large data has been a major concern for prediction systems like this. The objective is to easily use large real time data and create an effective user interface for better traffic management in a metropolitan dynamically evolving city like Bengaluru. This project aims at providing a localized more effective prediction system of handling traffic in the city.

II. SYSTEM DESCRIPTION

The system description of any system illustrates various components used. These components might be the hardware, or the software used by the system. In this paper, we have described the system from a hardware viewpoint, along with a software perspective. This system can be used by the end users as well as the traffic police department of that town/city.

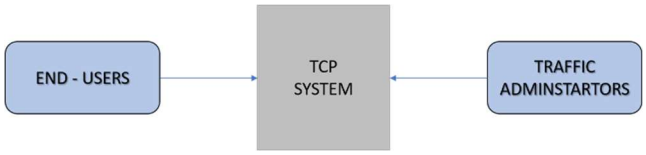

Fig 1: System Description

SYSTEM REQUIREMENTS

Non-Functional Requirements:

- Usability - The project is required to be easy to use and achieve its objectives of predicting the right crop will be grown more with high effectiveness.
- Reliability - The project must have requirements for the users to trust the system, even after using it for a long time. Its major goal must be to have a long mean time between failures.
- Accessibility - It must act as a universal design.
- Performance - The project must provide low response time at any point in time and under any circumstances, low utilization of computing resources and high accuracy.
- Scalability - The project must provide requirements to handle growing amount of work and accommodate it and provide scalability on the important dimensions like administrative, functional, geographic etc.

Hardware Requirements:

- Android Mobile (v6 and above)
- IR Sensor
- Arduino uno
- Battery

Software Requirements:

- Android Studio
- Windows System
- Cloud Services
- Blynk App

III. System Design

Architecture

The Traffic Congestion Model(TCP) consists of IR sensors, Arduino Uno , a Personal Computer, and mobile phones. The IR sensors detect possible congestion based on a variety of factors such as proximity, image recognition etc. The embedded system developed using Arduino stores and programs the raw data which are then transferred for further study to a local computer. This raw data is then sent to a PC for prediction and cleaning of the data. To predict congestion, a machine learning clustering algorithm such as the K-Nearest Neighbors (KNN) clustering algorithm is employed. This is sent to the cloud in Blynk where the result is stored. An application is built using an open source IOT framework to create a data-to-user interface that uses the outcome to simulate prediction

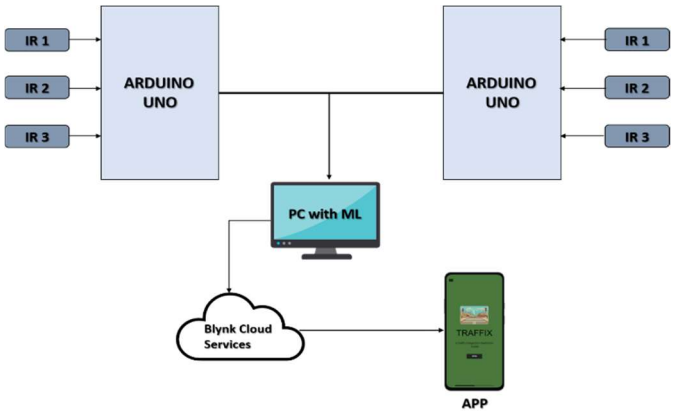

Fig 2: System Design

The system architecture consists of hardware components like:

- Infra-Red Sensors
- Alcohol Sensors
- An Arduino Uno/Nano Microcontroller Kit
- Blynk Cloud Services
- A computer with Machine Learning capabilities
- An android app called TRAFFIX

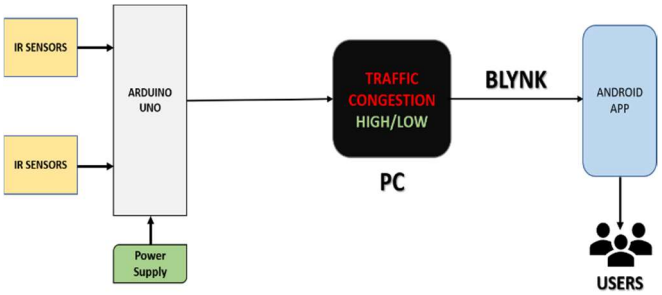

Fig 2: System Architecture

Software

The essential software's used are Android Studio, Windows System, Cloud Services , Blynk App. Other software requirements of the system include Keil IDE compiler, Android OS that includes Android SDK, Eclipse SDK and Keil Microvision 3 . All the components are used to fulfil the structure of software of the TCP model.

Hardware

The system architecture consists of hardware components like Infra-Red Sensors ,Alcohol Sensors, An Arduino Uno/Nano Microcontroller Kit, Blynk Cloud Services, a computer with Machine Learning capabilities and an android app called Traffix.

The hardware components include:

IR Sensors

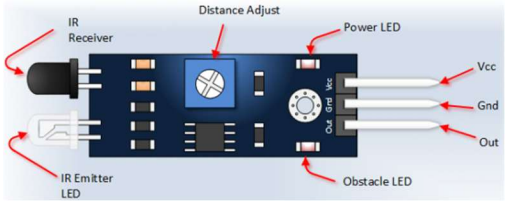

Fig 3: Infra-red Sensors

An infrared sensor is an electronic gadget, that produces to detect certain factors of the surrounding environment. An IR sensor can quantify the heat of the object just as it identifies its movement. This is put on paths and streets of a town to identify any movement by vehicles bringing about traffic jams.

Arduino Uno

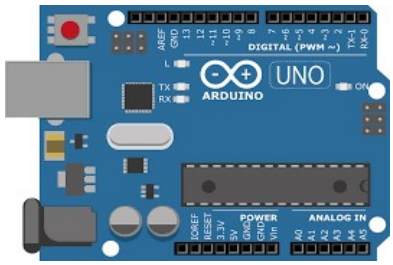

Fig 4: Arduino Uno

The Arduino Uno is a microcontroller that contains everything expected to assist the microcontroller; essentially interface it to a PC with a USB cable or power it with an AC-to-DC connector or battery to begin. We use Arduino uno to connect the IR sensors to the computer with machine learning capabilities. This is a very crucial components which acts as a medium between the hardware and the software.

IV. System Testing & Validation

The system is validated with Unit and Integrated testing of its various components.

Unit Testing

TEST UNIT	TEST CASE	RESULT
IR sensors are taken	They are interconnected to an Arduino Board	Through the sensors, the microcontroller gets the raw data for further analysis

Table 1: Data Collection Tests

TEST UNIT	TEST CASE	RESULT
Data is fed into ML system	Process the data the collected from sensors	The ML algorithm process the data and analyze for further decision
Algorithm process the traffic flow	Send the output in binary	The output is in binary and sent to next step for user output

Table 2: ML Module Tests

TEST UNIT	TEST CASE	RESULT
Congestion prediction display	Display the result of ML module for users	Result is displayed based the congestion prediction by the ML algorithm
Traffic prediction for each lane	Display the different traffic congestion level of each lane	The app output the traffic flow of each lane

Table 3: Blynk Module Tests

Integrated Testing

TEST UNIT	TEST CASE	RESULT
Data is collected from IR sensors	IR sensors are connected to microcontroller	The data is transferred to next module for analysis
Arduino board collects the raw data	Analyze the data	Data is analyzed and fed to ML
ML system uses algorithm to process the analyzed data	It tries with different algorithm and chooses the best fit	Algorithm outputs the data for visual output
The congestion is prediction	With given data prediction of congestion is predicted	The congestion is predicted and shown in Blynk app
Based of the congestion the system sends the output to user about the congestion	The system thus checks with the ml algorithm and data sets	The result of the congestion is then displayed

Table 4: Integrated Tests

V. Survey of similar products

Product Survey

Google Maps:

Google Maps is a web mapping platform created by Google. It offers satellite symbolism, airborne photography, road maps, 360° all-encompassing perspectives on boulevards (Street View), ongoing traffic conditions, and course planning for going by foot, vehicle, bike, or open transportation. In 2007, Google started offering traffic information as a hued overlay on the head of streets and motorways to speak to the speed of traffic. Officially supporting is utilized to get the GPS-decided areas of an enormous number of cellphone clients, from which live traffic maps are created. Google has expressed that the speed and area data it gathers to compute traffic conditions is mysterious. Choices accessible in each telephone's settings permit clients not to impart data about their area to Google Maps.

Waze:

Waze is a GPS route programming application owned by Google. It takes a shot at cell phones and tablet PCs that have GPS support. It gives turn-by-turn route data and client submitted travel times and course subtleties while downloading area subordinate data over a cell phone organize. Concerns have been communicated that the application situated on cell phones can be utilized to screen developments by recognizable people. Some street security advocates have voiced worry over the possibility of more drivers utilizing Waze, which they state can possibly divert them with a whirlwind of symbols and notices and put them at more serious danger of a mishap.

Yandex Maps:

Yandex Maps is a Russian web planning administration created by Yandex. The administration gives point by point guides of the entire world. It incorporates a pursuit, data about automobile overloads, directing, and road displays. Yandex at first had maps just of Russia and Ukraine, yet now incorporates world guides. Guides are accessible in four variants: maps, satellite pictures, satellite pictures with inscriptions, and legend (cross breed). Clients can quantify separate and get bearings by hauling between spots on a guide. The pursuit is accessible by both topographical articles (addresses, boulevards, urban communities, locales, and nations) and by associations. The brightening of the region, city, or locale is accessible subsequent to looking through the association on the site

Fig 5: Google Maps / Waze / Yandex Maps

VI. Related Work

Network-based traffic congestion notification service (US6253146B1)[26]

A server platform in the PSTN performs the service of calling the subscriber if the subscriber's customary commuting route is congested. The service makes an initial call when the fastest commute route is congested. It also makes additional calls each time it finds a better alternate route. The server can accommodate any subscriber commuting pattern. If a street on the subscriber's customary route is found to be congested, then the server sequentially tests additional streets having geographic coordinates nearby, dynamically searching for an alternate route that has an acceptable level of congestion. The server has a voice response unit (VRU) in the platform, to announce to the subscriber that the subscriber's customary commuting route is congested. Following an initial call to the subscriber, the server continues to search for better alternate routes having the least congestion and it calls the subscriber again if it finds a better alternate route.

Method and apparatus for traffic incident detection (US6411328B1) [26]

Contraption and strategies are powerful for recognizing, finding, and portraying traffic-related episodes depend on improved picture preparing methods applied to infrared and noticeable light range street pictures in a period succession. Significantly constant seclusion and distinguishing proof of atypical or startling traffic conditions permit control of traffic signals and dispatch of law authorization, upkeep, or crisis clinical consideration assets to lessen cost and increment security. Perceivable traffic-related occurrences incorporate, for instance, the presence of a fixed item on a street, the presence of a person on foot on a street, and the recognizable proof and area of vehicles blocking traffic stream by moving too gradually or sporadically or off course. Programmable advanced PC picture preparing permits programmed arrangement of the seriousness of recognized occurrences dependent on the potential for additional traffic dangers and wounds and can bolster programmed signs to change traffic stream designs for decreasing the probability of episodes, just as programmed cautions to law requirement, upkeep, and crisis clinical staff.

Systems and methods for detecting road congestion and incidents in real time (US9240123B2) [26]

Apparatuses and methods are provided for determining real-time traffic conditions. A candidate road is divided into road segments by perpendicular bisectors. A spatial sliding window is positioned over at least a portion of a road segment, wherein the spatial sliding window corresponds to a front end of the road segment in a direction of travel of the road segment. Real-time probe data is received from mobile devices in probe vehicles or on travelers of the at least portion of the road segment within the spatial sliding window. The real-time probe data is analyzed, and a computer program assists in determining the real-time traffic conditions of the at least portion of the road segment within the spatial sliding window. Based on the analysis, real-time traffic conditions are reported.

VII. Comparitive Analysis

Comparative analysis responds to inquiries concerning how and why a framework will respond to annoyances of its parameters. This paper shows how points of view can be utilized for near examination, sums up a sufficiency verification for the procedure, exhibits deficiency depicts a working usage, and presents exploratory outcomes. During the analysis, we compare our strategy for breaking down and predicting traffic to other preexisting strategies which use Machine Learning. In this paper, we look at two techniques chose on the premise that they use KNN Algorithm for predicting traffic congestion.

One such method that uses KNN algorithm is mentioned in the paper "*Traffic Congestion Prediction System using K-Nearest Neighbor Algorithm by Rishab Menon R, Shreyas M S, Rajashree P, Rohith Thammaiah*"[12] published in the International Research Journal of Engineering and Technology (IRJET)[25]. The method they suggest is to consider both temporal and spatial correlations while predicting the traffic. When KNN is implemented, their project predicts the traffic congestion values for the required day and time interval. Their strategy is to utilize the spatial, as well as the temporal relationships found in the rush hour gridlock dataset, so as to foresee the traffic clogs at a necessary area, on a specific day and timespan. In this instrument, the traffic at a street is subject to the traffic at its neighboring streets from where the traffic can stream into the street under consideration. Along these lines, the neighbor's traffic information should likewise be considered to display the traffic on this street. As it were, the spatial relationships are considered alongside the temporal connections with the assistance of the KNN algorithm to predict traffic congestions.

Whereas

Our traffic congestion prediction technique is to deploy live Infra-red sensors on specific lanes to record real-time data. By utilizing these sensors and the information we get for processing, yields an accurate measure of the congestion occurring at that road/street. Due to constant monitoring of the data, we can get, the absolute minimum estimation of congestion just as indisputably the maximum measure. This minima and maxima values cause our model to examine the traffic density at a specific area effectively with extreme precision. The commonality between the 2 strategies i.e. the KNN algorithm is used by our model to clean the data , process the data, execute it with as many iterations possible to get the desired value of congestion which is then used by Blynk IoT to predict congestion. Finally, we project the output onto the mobile application for effective user interaction with the model and its features.

VIII. IMPLEMENTATION

HARDWARE CIRCUIT

-

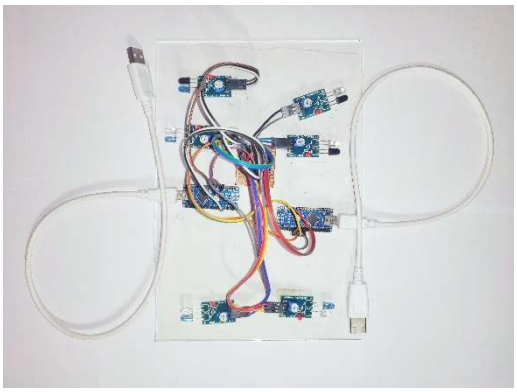

Fig 6: Hardware Circuit

The hardware circuit (as shown in **Fig 6**) consists of :-

- 4 Infra-Red sensors
- 1 Breadboard
- 2 Arduino Uno Microcontrollers
- Insulated copper wires
- USB cables

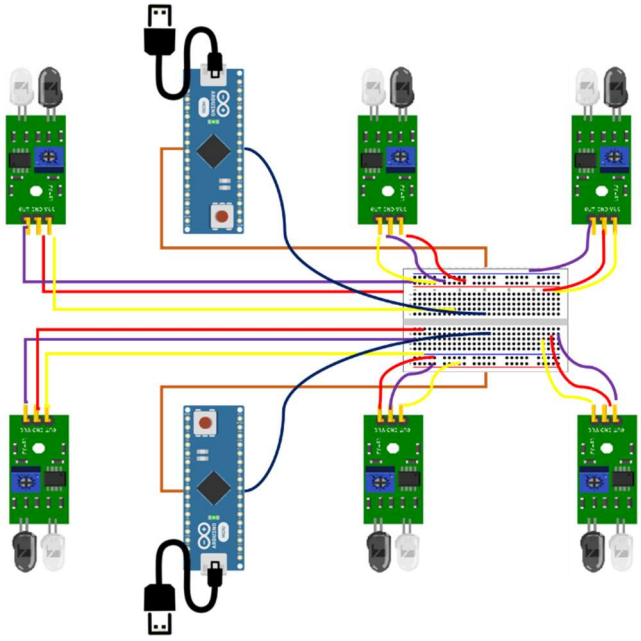

Fig 7: Pictorial representation of the circuit

The IR sensors recognize conceivable blockage dependent on an assortment of variables, for example, vicinity, picture acknowledgment and so forth. The installed framework created utilizing Arduino stores and projects the crude information which is then moved for additional examination to a neighborhood PC. This crude information is at that point sent to a PC for forecast and cleaning of the information. To foresee blockage, a machine getting the hang of bunching calculation, for example, the KNN grouping calculation is utilized. This is sent to the cloud in Blynk where the outcome is put away. An application is assembled utilizing an open-source IoT system to make an information to-UI that utilizes the result to recreate expectation.

BLYNK IOT [27]

Blynk is a toolset for all producers, innovators, designers, educators, who might need to utilize their cell phones to control hardware like Arduino, Raspberry Pi, and comparative ones. With Blynk, we can essentially assemble an astounding interface from different gadgets that are given, transfer our code to the equipment, and get brings about no time. It works perfectly for the individuals who are getting into creating. Blynk works with all sheets and shields. The Blynk cloud administrations are free and open source. Utilizing Blynk is simple as envisioning a prototyping board on a cell phone where we can relocate catches, sliders, showcases, charts, and other utilitarian gadgets, and through that, we can control the Arduino board utilizing cell phones. Blynk is not an application that works just with a shield. Rather, it has been intended to help the sheets and shields you are as of now utilizing. Also, it takes a shot at iOS and Android. Regardless of what sort of association we pick - Ethernet, Wi-Fi, or ESP8266, Blynk libraries and demo representations will cause the client to go on the web, interface with Blynk Server and pair up with a cell phone one after another.

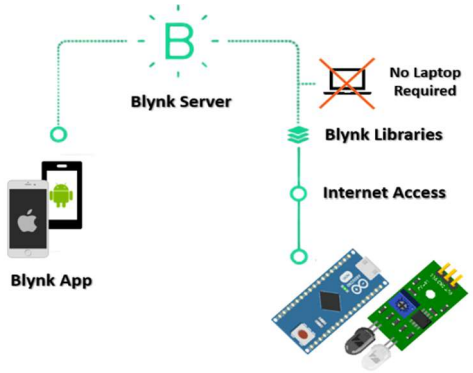

Fig 8: Blynk Architecture

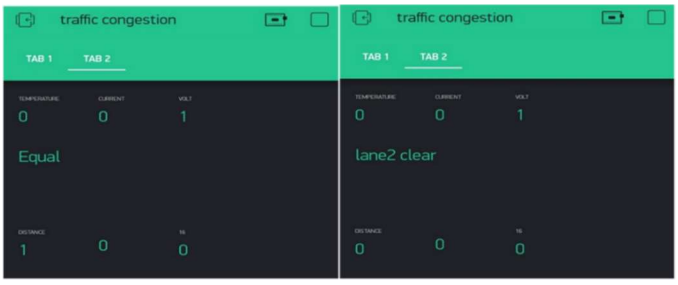

Fig 9: Blynk TCP Results

ANDROID APPLICATION

The android application is named "TRAFFIX" and is made utilizing Android Studio. The fundamental adaptation of the application has a landing page, menu page which records the paths/streets of that area, singular pages for every path.

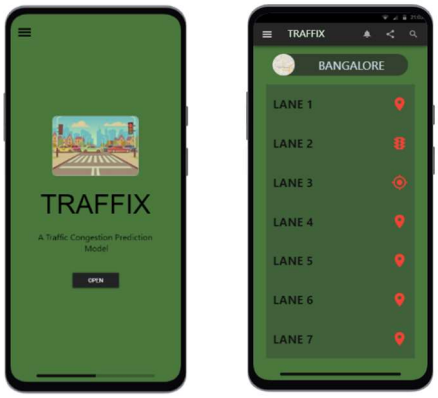

Fig 9: Home page and Lanes Page

At the point when the client taps on any path recorded on the application, the client is taken to that individual path page. The page portrays the path and there is a choice to know the traffic status of that path. For instance, when a client needs to know the status of Lane 1, he can essentially go to that page and tap on the status button. On clicking, it shows whether the lane is congested or clear which is dependent on the output it gets from the KNN algorithm.

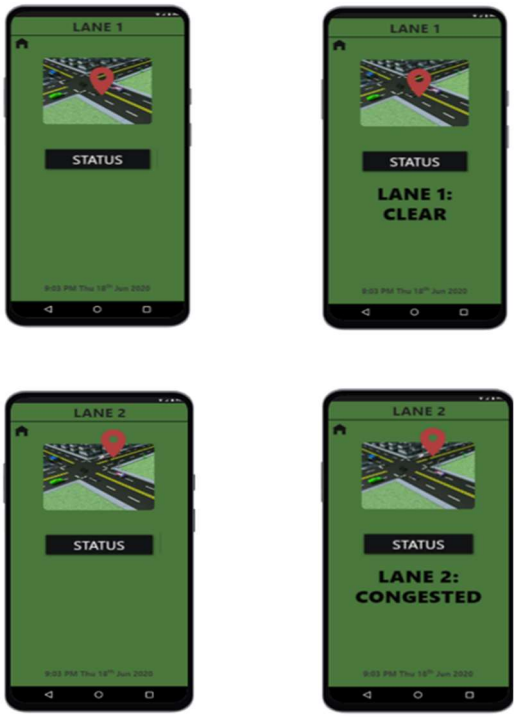

Fig 10: Lane 1 and Lane 2

K-NEAREST NEIGHBOR ALGORITHM

KNN algorithm is applied on the input data received from the Arduino microcontroller. The algorithm considers the neighbor nodes dependent on separation between the nodes and time alone. The predicted values are projected on to the android app.

Fig 11: KNN Algorithm

IX. CONCLUSION

The proposed Machine Learning Congestion Prediction calculation, which utilizes the K-Nearest Neighbor algorithm, offers a speedy, solid, and early estimate of traffic congestion for a given static road network that can be known as a graph. The multifaceted nature of the algorithm alluded to above would be consistent. This can be adequately utilized by any gadget with less processing capacity and fewer assets. As an expected route, the forecast of traffic congestion can be predicted by utilizing a scope of hybrid procedures that can create a high precision performance. This project can be executed in urban communities with a lower traffic density, and this can be viewed as the base of a forecast model, while the outcomes for a city like Bengaluru can be taken as the most extreme to locate the viable estimation of this expectation model. Besides, this can be changed into state-supported urban transport management and can be utilized by the Traffic Police Department alongside residents living in the city. The task is viewed as localized to cater to the individuals living in the city. Therefore, further redesigns and highlights can contend with goliaths like Google Maps, Waze, and Yandex Maps.

X. ACKNOWLEDGEMENT

The authors thank and gratefully acknowledge the support of Prof. Shanthi D.L (BMS Institute of Technology and Management, Bangalore).

XI. REFERENCES

[1] Lippi M, Bertini M, Frasconi P, "Short-term traffic flow forecasting: experimental time-series analysis and supervised learning," IEEE Trans. Intelligent transport systems, vol. 14, 2, pp. 871-882, 2013.

[2] Mr. Tomizuka, "Automated Highway Systems – A Next Century Intelligent Transportation System," IEEE / ASME International Conference on Advanced Mechatronics, Japan, 1997.

[3] Bob McQueen and Judy McQueen, Intelligent Transportation Systems Architectures, 1999.

[4] S. Ghosh and T. Lee, Intelligent Transportation Systems – New Principles and Architectures, CRC Press, 2000.

[5] Nakashita H., Okamoto H., Kawabat T. "Advanced Mobile Traffic Information and Communication System (AMTICS)," Meeting of the Transportation Research Board, 1988.

[6] Chen X.M, Li, L; Stochastic Evolutions of Dynamic Traffic Flow Modeling and Applications, Chapter 2 – Springer 2015.

[7] K. Q. Weinberger, L. K. Saul, Distance metric learning for a large margin nearest neighbor classification, Journal of Machine Learning Research.

[8] H. Chang, D.-Y. Yeung, Kernel distance metric learning for content-based image retrieval, Image and Vision Computing 25 (5) (2007) 695–703.

[9] Machine Learning based traffic congestion prediction in IoT-based Smart City by Suguna Devi, T. Neetha.

[10] Traffic-Congestion Prediction Algorithm Based on Pheromone Contact Model by Satoshi Kurihara.

[11] Improved K-nearest neighbor model of short-term traffic flow prediction by Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei.

[12] Traffic Prediction System using the KNN Algorithm by Rishab Menon, Shreyas M S (2020).

[13] Traffic congestion anomaly detection and prediction using deep learning by Adriana-Simona Mihaitaa, Haowen Lib and Marian-Andrei (2020)

[14] "Internet of things for smart cities" by A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, IEEE Internet of Things journal. (2014)

[15] City-Wide Traffic Congestion Prediction Based on CNN and LSTM by Navin Ranjan , Sovit Bhandari, Hong Ping Zhao, Hoon Kim, And Pervez Khan. (2020)

[16] Leveraging Personal Navigation Assistant Systems Using Automated Social Media Traffic Reporting by Xiangpeng Wan, Hakim Ghazzai, and Yehia Massoud. (2020)

[17] Real-Time Navigation in Urban Areas Using Mobile Crowd-Sourced Data Xiangpeng Wan, Hakim Ghazzai, and Yehia Massoud. (2019)

[18] Traffic-Sign Classification Using Machine Learning Concepts by A. Obulesh, P. Sri Sahithi, S. Pavitra (2020).

[19] Traffic Clustering Algorithm of Urban Data Brain Based on a Hybrid-Augmented Architecture of Quantum Annealing by Ning Wang, Gege Guo, Baonan Wang, and Chao Wang. (2020)

[20] Region-wide congestion prediction and control using deep learning by Sudatta Mohanty, Alexey Pozdnukhovb, Michael Cassidyc. (2020)

[21] Traffic Congestion Identification and Reduction by Brijesh Kumar Chaurasia1, Wellington S. Manjoro, · Mradul Dhakar. (2020).

[22] Optimal Seeds Discovery of Traffic Congestions by Carlos Bermejo, Ting Wu, Xiang Su, Pan Hui. (2020)

[23] Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment by Amr Elfar, Alireza Talebpour, and Hani S. Mahmassani. (2018)

[24] Institute of Electrical and Electronics Engineers (IEEE)

[25] International Research Journal of Engineering and Technology (IRJET)

[26] Google Scholar and Patents

[27] Blynk IoT