Pollution Based Intelligent Navigation System Using Dijkstra's Algorithm

Sujatha Rajkumar[1], Dhruv Haribhakti[2], Shreyas Loya[3], Sivakumar Subramaniam[4]

[1]Assosciate Professor, Vellore Institute of Technology, Vellore, India.
[2]Student, Vellore Institute of Technology, Vellore, India.
[3]Student, Vellore Institute of Technology, Vellore, India.
[4] Senior Lecturer, Universiti Teknikal Malaysia, Melaka, Malaysia.

[1]sujatha.r@vit.ac.in, [2]dhruvharibhakti7@gmail.com,
[3]shreyasloyaloya@gmail.com, [4]siva@utem.edu.my

Abstract

The Global Positioning System or GPS has become an essential part of our lives today. Using Google Maps, people can check their live location as well as find out the minimum distance that they need to travel in order to reach their destination. However, pollution is on the rise and today it has become a major goal for us to find out a way to use alternative methods in order to make sure that we limit our exposure to the same. In this research work, a prototype model will be made that will continuously send pollution levels in its vicinity to the cloud and in turn to an application. The application can be interfaced with Google Maps which uses the GPS navigation system. In order to achieve this, pollution monitoring sensors will be placed in different locations which will continuously send the pollution data to the cloud and from the cloud to the application. The main objective of this research paper is to suggest the least polluted path the user can travel on in order to reach his/her destination. The least distance between the source and the destination can also be taken as a factor and an optimum route can be suggested to reach the objective by taking the distance as well as the pollution levels into account.

Keywords: *Internet of Things, Gas sensor, Dijkstra's Algorithm, Navigation, Pollution*

1. Introduction

GPS or Global Positioning System is a navigation system where information is supplied by satellites which are owned by the US government. It provides geolocation and time-based information anywhere on Earth. This system only works if there is a direct Line of Sight (LoS) from the location to at least four GPS satellites. Google Maps is an application that uses this GPS technology to show us the shortest distance between two locations.

Today, air pollution is constantly on the rise. The burning of fossil fuels, the evacuation of exhaust from industries, agricultural activities, vehicle emissions, mining operations, etc. all contribute to the increasing air pollution. Due to this, the world is facing major catastrophes like global warming and ozone layer depletion [7]. Along with that, respiratory and heart diseases, acid rains, eutrophication, and natural disasters also follow.

Studies and statistics show us that 'Vehicles' are the major contributing factors towards climate change comprising a whopping 41% of the total air pollution sources [6]. People can't be asked to stop using cars and other vehicles altogether since they are the most important means of road transport. Also, everyone cannot suddenly shift to electric or hybrid cars since they are extremely expensive and have not evolved completely. Also, factories and industries cannot be asked to stop functioning entirely. One thing that can be done to help is re-route the paths of the vehicles from highly polluted areas to areas having lesser pollution. This will not help to eradicate pollution, but it will help in balancing it until the point when some permanent solutions to the problem at hand can be found.

Finding alternate routes for the vehicles will ensure that the pollution levels in the more densely polluted areas will not get more polluted because vehicles travelling through those areas will be reduced to a minimum. The area through which the vehicles will be asked to travel will also have a certain threshold value for the highest pollution level allowed which our prototype model will ensure that it is not exceeded. Gradually, as the air becomes clearer in the densely polluted areas, vehicles will again be allowed to travel through the same.

2. Literature Review

People have already tried to implement ways and methods to control the pollution being emitted from vehicles every day. One paper published in 2017 implemented a system where the emissions being emitted from the exhaust chambers of the cars are detected using semi-conductors. As soon as the pollution or emission level shoots above the previously set threshold level, a message is sent to the driver and he/she is fined a certain amount of money [8]. The pollution detector is connected to the cloud storage and the data is stored using the cloud database services. Hadoop tool is used to retrieve data from the cloud [1, 3]. The authors of a certain paper published in 2019 used Azure cloud services for agricultural applications [5]. One of the most important factors to be considered while using cloud services is the data security which has been discussed in "Optimal ECC Based Hybrid Signcryption for MANET Security" [2]. Another paper implemented a low-cost air quality and pollution monitoring system that combined a small-sized, minimum-cost sensor to an Arduino microcontroller unit [10, 11]. The processing software was able to analyze the collected quality data with high precision [9].

There is also an application known as 'Airlief' [4] which provides air quality data in more than 80 countries worldwide. Airlief gives its users the tools necessary to find out the amount of pollution in the areas of our choice. It also colour codes the pollution levels based on the concentration of the gases. However, Airlief still does not map the route from a point to another. The least polluted path cannot be chosen by using Airlief. A research paper published in 2016 also mentioned some ways to handles smart data which we could learn from [14]. Research papers published in 2018 successfully used the NodeMCU ESP8266 module in order to make a fully functional monitoring system and a home automation system [17, 18]. This inspired us to use the same.

In a paper published in 2017, the authors came up with a system to manage the traffic in a city. The authors used Dijkstra's algorithm to find the path having less traffic. This model can be useful for creating city plans, but for large distances, it is not beneficial to bring down the average pollution of the entire city. The city still remains equally polluted [12].

A few authors tried to implement a system where in a city bus keeps giving them real-time pollution readings in its vicinity while it moves around the city [13]. Two papers published in 2018 explained in detail how to make applications by taking real-time data from Google Firebase [15,16]. Also, a few research papers managed to use different clouds ranging from Node-RED to AWS IoT to cloud platform in order to make their respective models which were sensor integration models and remote monitoring systems [19-22]. Reading about these prototypes helped us in selecting a suitable cloud for our research.

3. Objective

It is very important to keep a proper balance in each and every system in our ecosystem. These days there are cities which have extreme pollution levels. This leads to an improper balance of pollution in the ecosystem. This results in the varied heating up of the regions and the creation of unnatural temperature belts in the atmosphere. In the end, sudden climate changes and extreme weather are experienced.

The main objective of this research work is to maintain this balance of pollution in the atmosphere. The cities with very high pollution levels will be largely benefited from this model. The pollution levels of these cities will decrease a considerable amount and the pollution level will get balanced.

4. Proposed Method

The primary aim is to navigate and choose the path which has the minimum amount of pollution in the air. It is similar to the navigation of Google maps but instead of coming up with the shortest route the application will provide the path with minimum pollution level.

From the vast array of air quality monitoring sensors, an accurate sensor is selected that matches the problem taken and provides real time data. These sensors can then be placed throughout an area. To identify the shortest distance between 2 points (the source and the destination), Google uses the data from mobile phones connected to the internet and provides information based on traffic congestion. With the help of sensor data, the user will be guided to navigate through a path having the minimum pollution. All the data will be processed using a Node MCU board and will then be sent to an application which in turn will take this data as input and will provide the best path i.e. the least polluted path as output.

This model will work on the basis of Dijkstra's Algorithm by taking data obtained from sensors into account and initializing them as weights.

5. Dijkstra's Algorithm

Dijkstra's Algorithm is the algorithm used to identify a path in a graph with the least possible cost between the nodes. The algorithm consists of a source node and a destination node. The cost of each edge between every two nodes is pre-defined. The algorithm starts from the source node checks all the possible ways and then chooses the path with the least cost. Dijkstra's Algorithm can be used to analyze the shortest distance between two cities. For example, cities can be replaced with nodes and distance between two cities can be replaced with the cost between two nodes in the graph. Google Maps uses its own shortest path finding algorithm for Navigation. It considers the parameters such as traffic and tolls between two cities along with the distance. So, when the user defines the source city and the destination city, the app can find the shortest and least time-consuming route for the user. In our model, an air pollution parameter will be added to the cost in order to find the route not only which is less time consuming but is also least polluted.

5.1. Shortest Path Algorithms

Considering the shortest path problem, three algorithms namely Bellman-Ford, Floyd-Warshall, and Dijkstra Algorithms can be applied. The prototype requires us to use a single-source shortest-path approach, where the shortest path from a single point to all other points in a graph is found. The Floyd-Warshall Algorithm uses all-pairs shortest-path approach, which would help to find the shortest path between all pairs of points in a graph. Hence, Floyd-Warshall Algorithm would not be useful for our prototype because of its high complexity.

Bellman-Ford and Dijkstra's Algorithm, both are used to find single-source shortest-path in the given graph. Bellman-Ford's Algorithm can work when there is a negative weight edge and is used to detect the negative weight cycle. It also uses Dynamic Programming to

ISSN: XXXX-XXXX IJXX

find the shortest path. Dijkstra's Algorithm cannot work when there is a negative weight edge. It uses the Greedy approach to find the shortest distance.

According to our scenario, all the weighted edges of the graph will be positive as neither the distance nor pollution values can be negative. In Bellman-Ford's Algorithm, the total distance between source and destination can be reduced after subsequent updates because of the inclusive negative-weights. While in Dijkstra's Algorithm, the total distance is the actual length of the shortest path. Hence, to find the accurate distance between the source and the destination node, Dijkstra's Algorithm can prove to be more accurate and requires lesser time to implement, considering the worst-case scenario, as compared to Floyd-Warshall and Bellman-Ford Algorithms.

Table 1. Computational Complexities of shortest-path algorithms

Sr. No.	Shortest path Algorithms	Computational Complexity
1.	Bellman-Ford-Moore algorithm	$O(n\bullet m)$
2.	D'Esopo-Pape algorithm	$O(n\bullet 2^n)$
3.	Pallottino's algorithm	$O(n^2\bullet m)$
4.	Goldfarb-Hao-Kai algorithm	$O(n\bullet m)$
5.	Goldberg-Radzik algorithm	$O(n\bullet m)$
6.	Dijkstra's algorithm	$O(m\bullet \log n)$

In one of the papers published in 2018, the authors have discussed various shortest-path algorithms [13]. The Bellman-Ford-Moore algorithm stores all the vertices in a queue in first-in first-out format for labelling. The labelled vertices are added at the tail and vertices ready to be scanned are removed from the head of the queue. The D'Esopo-Pape algorithm uses a deque for labelling the vertices. The labelled vertices are added at the tail, for first time and at the head, otherwise. The vertices ready to be scanned are removed from the head of the queue. Pallottino's algorithm works with two FIFO queues, one for labelled vertices and other re-labelling vertices.

The Goldfarb-Hao-Kai algorithm uses the breadth-first-search algorithm with one queue and the Goldberg-Radzik algorithm uses topological-scan algorithm with two queues.

5.2. Implementation of the Algorithm

Fig. 1 illustrates the overall working and implementation of Dijkstra's Algorithm in six steps. As shown in Fig. 1.1, the cost for each node is set as ∞. Following that, the costs for all the neighboring nodes of A are found as shown in Fig 1.2. The costs are found to be 14, 9 and 7 for nodes 2, 3 and 4 respectively. As the cost for node 4 is the least, so it is approached first. In Fig. 1.3, the total costs for all the neighboring nodes of node 4 are to be found. The costs are found to be 17 and 22 for nodes 3 and 4, respectively. As the previous cost for node 3 is less than the current cost, therefore 9 is considered as the cost for node 3. Thereafter, the focus moves towards node 3 and finds the total costs for all the neighboring nodes of the same. As seen in Fig. 1.4, the costs are found out to be 11 and 20 for the nodes 2 and 6. As the previous costs for nodes 2 and 6 are larger than their current costs, therefore 11 and 20 are considered as the new costs for nodes 2 and 6, respectively. Next node 2 is considered, since its cost is less than the cost for node 6, i.e. 11<20. After that, the only node left is B, which is the destination. Then the updated cost of B will be 20. Following that, node 6 needs to be considered, and the total cost for B from node 6 is found to be 26. In Fig. 1.5, the path having least cost for the destination B is to be decided. As 20<26, the path with lowest cost required to reach from A to B will be A→3→2→B. Fig. 1.6 illustrates the complete route with the lowest cost, which is highlighted with blue color.

ISSN: XXXX-XXXX IJXX

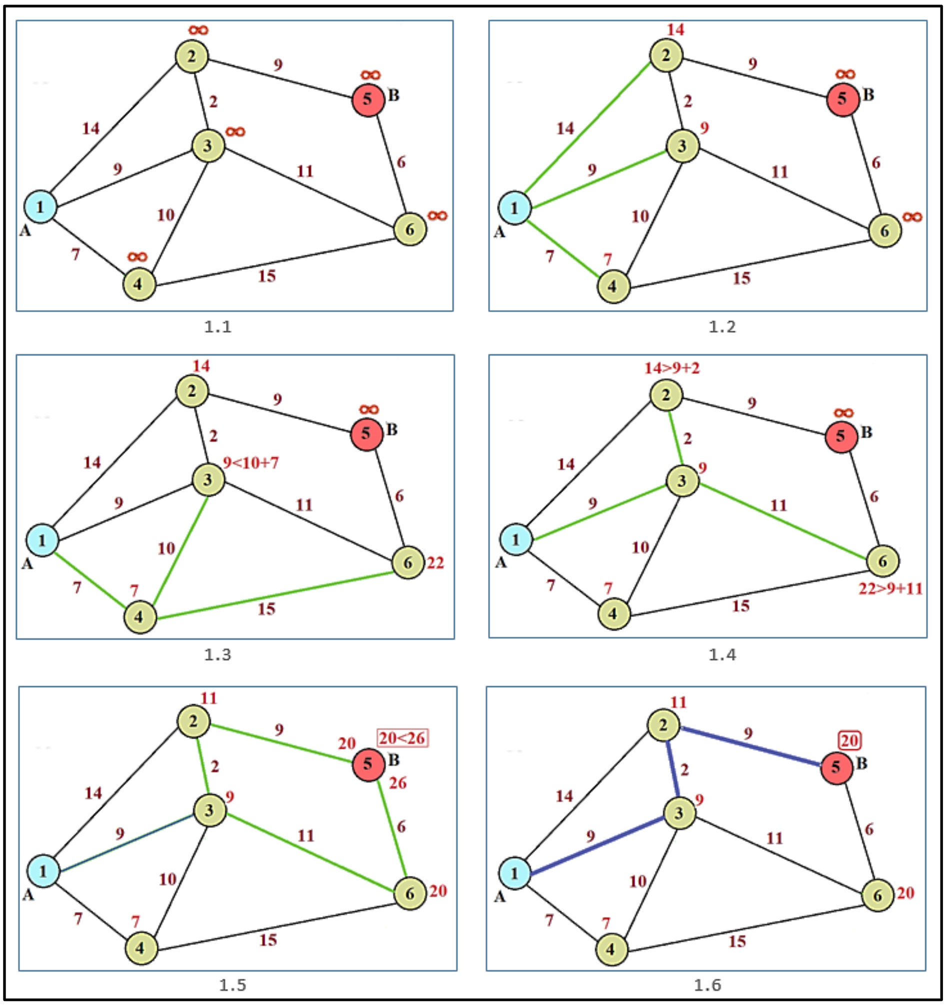

Figure 1. Figure Label

5.3. Computational Complexity

The simplest form of implementation can be done by storing vertices in an array or linked list. This will result in the time complexity of O(|V|^2 + |E|), where |E| is for search needed to find the minimum distance node. For sparse graphs, where there are very few edges and many nodes, it can be implemented more efficiently if the graph is stored in an adjacency list using a binary heap. This will produce a computational complexity of O((|E|+|V|).log|V|). Dijkstra's Algorithm identifies the shortest path to all nodes, using the Greedy approach. However, it is about as computationally expensive to calculate the shortest path from one node to all other nodes using Dijkstra's Algorithm as it is to calculate the shortest path to some particular vertex. Therefore, in order to know the best path from one source to some other vertex, Dijkstra's algorithm is used.

6. Process Flow

Fig. 2, as given above illustrates the process flow of our entire research work. Pollution data needs to be collected from various sensor nodes throughout the area. The number of iterations is defined as 'i' and the variable 'N (here, 6)' is assigned to the number of locations in total. Now data is collected from the MQ-7 sensors and is processed using a microcontroller. Data is then pushed to the cloud using a Wi-Fi module. The cloud carries out analytics on the data. A certain threshold is set above which, the air is considered as highly polluted and below it, the air is considered as safe to travel. The highly polluted air is color coded as Red and the slightly polluted but safe air is color coded as Green. An array POLL[] is created consisting of all the areas where our sensor nodes are deployed. It will have 'N' number of values in it. Now, the ppm value of the first sensor is taken and divided by the sum of the overall pollution ppm of all the N sensors combined (sum_poll). Once done, this value is multiplied by 100 in order to get the percentage value. The same process is carried out for all the N sensors. Similarly, the percentages of the distances between the nodes is calculated. The overall weights are then calculated for all the edges by taking 30% of the distance percentage between the nodes and 70% of the pollution percentage and adding them together. The percentage values are taken as 70% since it is important to give more preference to the pollution and the distance values are given 30% weightage since it is important that shortest route possible from the source to the destination is calculated. Now, the overall weights calculated are defined as WEIGHT[i], where 'i' represents the nodes. Now, if i<N, then the process is started again from the step of data collection by the sensors. If not, then the source node is defined as NODE1 and is assigned the variable 'N1' while the destination node is defined as NODE2 and is assigned the variable 'N2'. Dijkstra's algorithm is then used to analyze the shortest route between N1 and N2 using WEIGHT[i]. Finally, the optimum path is displayed as the output.

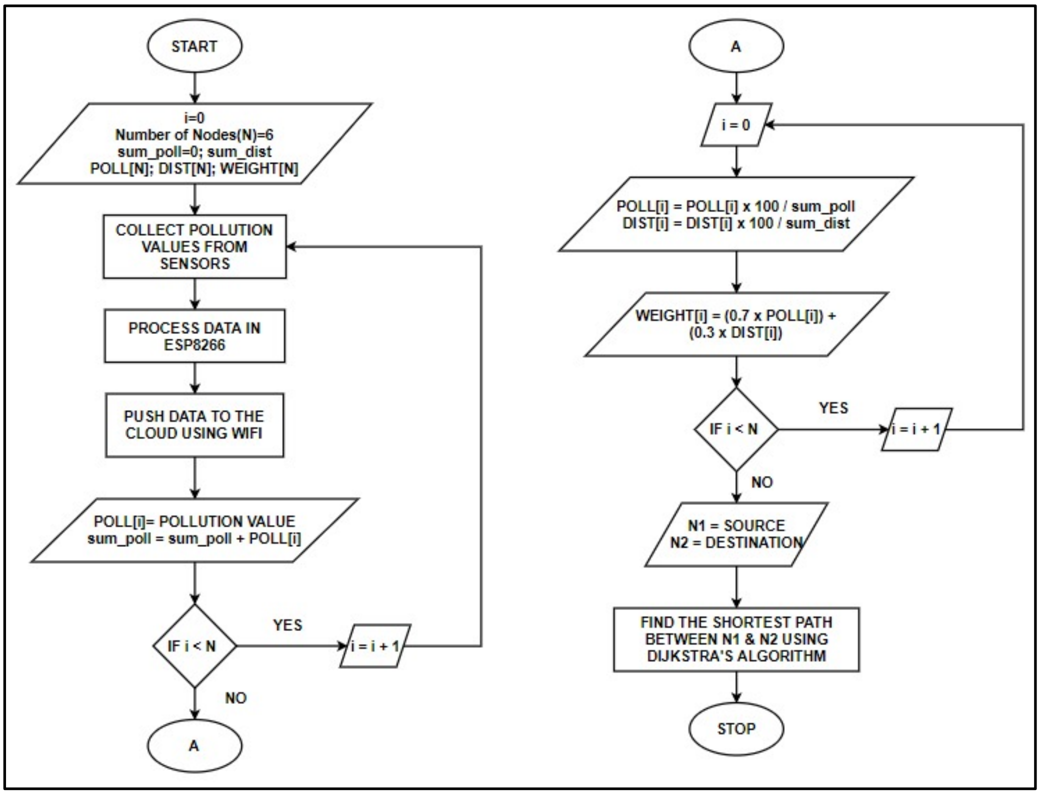

Figure 2. Process Flow

7. Methodology

Our proposed work will constitute of two parts- hardware and software. The hardware model will consist of a micro-controller, sensors, Wi-Fi module and power source. The power source will help to power the sensors and take the pollution values from the surroundings. It will be processed by the microcontroller and sent to the desired cloud with the help of Wi-Fi module. There will be hardware units installed in different parts of the

ISSN: XXXX-XXXX IJXX

cities, and the average pollution level for the city will be considered as the cost for that city. The real time values of pollution levels of all the available cities will be saved by sending these values to the cloud.

The software model will consist of a cloud service that can store the real time values of pollution levels of different cities. The cloud will generate unique API key for the database which can be read by the Mobile Application. The app will assign these pollution levels to the respective cities and will generate the cost between the cities at the time of navigation.

8. Results

After implementing the code on the ESP8266 (here, the NodeMCU Board), the sensors gave their outputs. This has been shown in Fig. 3.1. After the sensors gave their Carbon Monoxide output, Dijkstra's algorithm was calculated by taking the source node as Node 1 and the destination node as Node 5. The same graph and weights as shown in Fig. 1 have been considered. The shortest path (1→2→5) was found out and the distance to be covered was also shown (23).

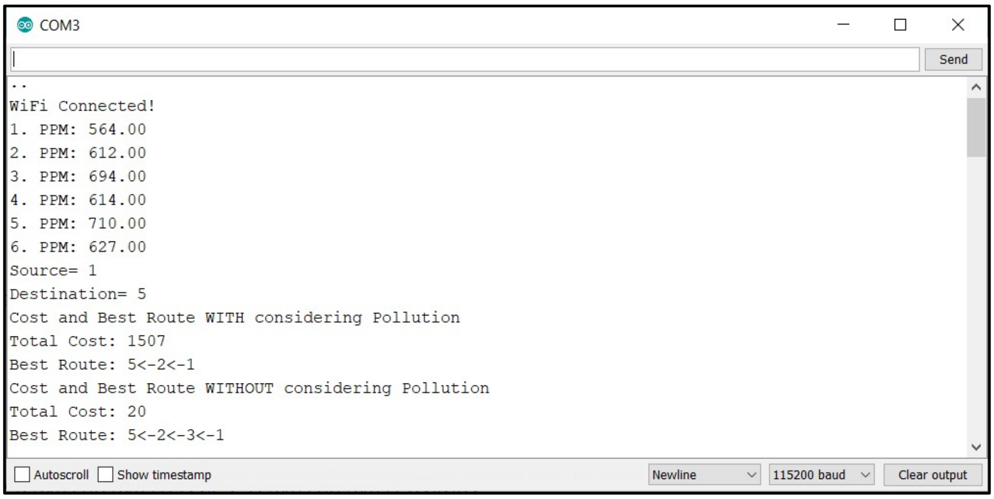

Figure 3.1. Serial Monitor before polluting Node 2

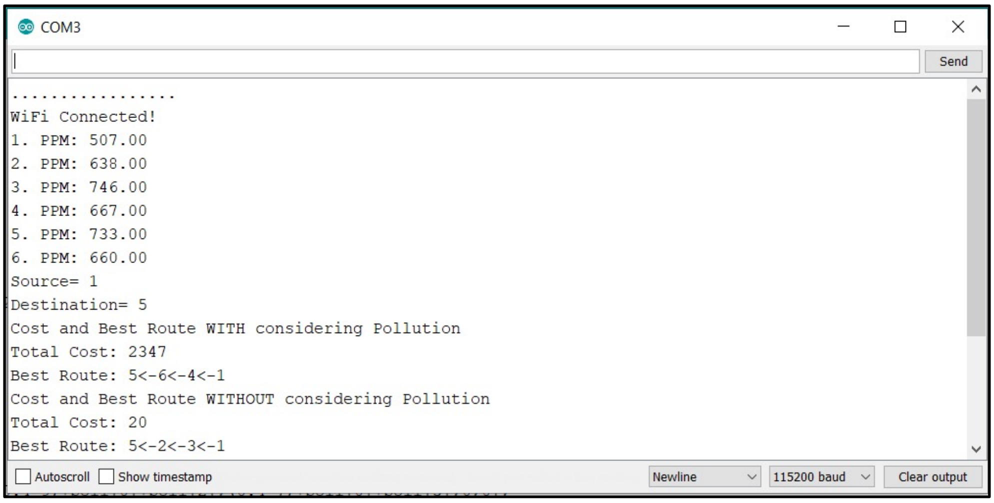

Figure 3.2. Serial Monitor after polluting Node 2

The Node 2 was then polluted and the new revised path (1→4→6→5) along with the new revised distance to be covered (20) was given as the output as shown in Fig. 3.2.

Finally, the data (the optimum path) was pushed from the cloud into the mobile application. The screenshots of the app screen can be seen in Figure 4. Figure 4.1 illustrates the optimum route in the mobile application before the Node 2 was polluted and Figure 4.2 shows the alternate route in the mobile application after the node was polluted. It can be observed how the shortest path changes in real-time when the pollution at a particular sensor node increases or decreases.

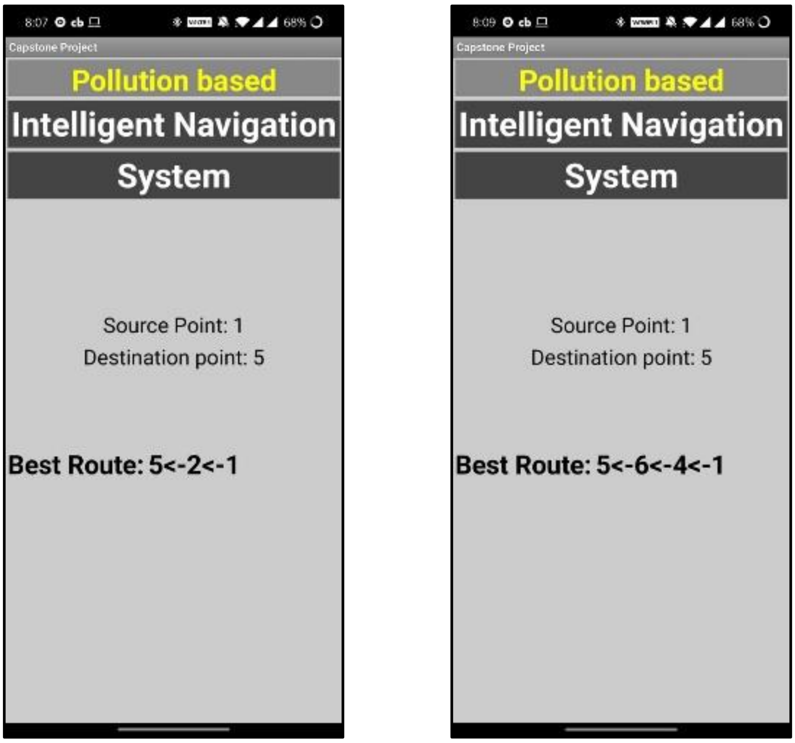

Figure 4. Mobile Application

9. Conclusion and Future Scope

The system proposed here displays the paths having the least pollution and the shortest routes where the pollution is given a priority (70%). Using this map, the user will be able to travel safely avoiding the areas having high amounts of air pollution and low air quality. The data fed into the maps in real time and the output of the optimum route is also calculated in real time so that the current situation of the roads can be taken into consideration.

The scope of improvement in this research work is that unique threshold levels for each city can be kept on the basis of its previous records. For the affected city, the re-routing process will continue until its real-time pollution level is above the threshold. Once it reaches the threshold, the re-routing process can be stopped. This will need the intervention of machine learning algorithms for finding the threshold of each city on the basis of its previous records. The algorithm will become very complex and is difficult to execute.

References

[1] V. Vitan, G. Berz, L. Saini, J. Arethens, B. Belabbas and P. Hotmar, "Research on alternative positioning navigation and timing in Europe," Integrated Communications, Navigation, Surveillance Conference (ICNS), Herndon, VA, 2018, pp. 4D2-1-4D2-17.

[2] Sujatha.R and Ramakrishnan.M, “Optimal ECC Based Hybrid Signcryption for MANET Security”, International Journal on Intelligent Electronics Systems, Vol. 10 No.1 January 2016.

[3] Abdelaziz El Fazziki, Djamal Benslimane, Abderrahmane Sadq, Jamal Ouarzazi ad Mohamed Sadgal, “An Agent Based Traffic Regulation System for the Roadside Air Quality Control”, IEEE, 2017.

[4] Sujatha Rajkumar, Arun.M, Jagriti Hirwani and Sirohi Sajal Sanjeev, “Predictive Analysis of Crops Cultivation for a Smart Green Environment Using Azure Services”, International Journal of Recent Technology and Engineering, Vol-7, Jan 2019.

[5] Homayon Zahmatkesh, Mohsen Saber and Majid Malekpour, “A new method for urban Travel route Planning based on Air pollution sensor data”, Current World Environment, 2014.

[6] Kingsy Grace. R., Manimegalai. R., Geetha Devasena. M.S., Rajathi. Si, Usha. K., Raabiathul Base-ria. N, “Air Pollution Analysis Using Enhanced K-Means Clustering Algorithm for Real Time Sensor Data”, Institute of Electrical and Electronics Engineers, 2016.

[7] Priyanka Phadtare, Shemant Kumar Singh, “Intelligent Navigation System using Air Quality Index”, International Research Journal of Engineering and Technology , 2018.

[8] Dr. L. Surendran, A. Suresh Kumar, S. Vanmathi, S. Sowbharraniga, “Cloud Based Vehicle Pollution Detection and Monitoring System”, International Journal of Advanced Research Trends in Engineering and Technology, 2017.

[9] Md. Abdullah Al Ahasan, Saumendu Roy, A. H. M. Saim, Rozina Akter, Md. Zakir Hossain, “Arduino-Based Real Time Air Quality and Pollution Monitoring System”, International Journal of Innovative Research in Computer Science and Technology, ISSN: 2347-5552, Volume-6, Issue-4, July 2018.

[10] Chaitanya, H. P. and H. Prasanna Kumar, “Automated System for Air Pollution Detection and Control of Speed in Vehicles”, International Journal of Advances in Engineering & Technology, 2016: 443.

[11] Ojekudo, Nathaniel Akpofure, Akpan, Nsikan Paul, “Application of Dijkstra’s Algorithm to shortest route problem”, International Organization of Scientific Research-Journal of Mathematics, Volume 13, Issue 3 Ver. 1, May-June 2017, PP 20-32.

[12] Amr Elmasry and Ahmed Shokry, “A new algorithm for the shortest-path problem”, Networks, Vol.-74, 21 December 2018.

[13] Sami Kaivonen, Edith C.-H. Ngai, “Real-Time Air Pollution Monitoring with sensors on city bus”, Digital Communications and Networks 6 (2020) 23-30.

[14] Ahlgren, M. Hidell, E. Ngai, “Internet of things for smart cities: interoperability and open data”, IEEE Internet Computing 20 (6) (2016) 52–56.

[15] Khawas, Chunnu & Shah, Pritam, “Application of Firebase in Android App Development-A Study.” International Journal of Computer Applications, 2018. 179. 49-53.

[16] Chatterjee, Nilanjan & Chakraborty, Souvik & Decosta, Aakash & Nath, Asoke. “Real-time Communication Application Based on Android Using Google Firebase”, International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS), 2018.

[17] Aziz, Dlnya, “Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module”, International Journal of Scientific and Engineering Research, 2018. 9. 801.

[18] Mohammad Miraj Shekh, Asha S. R., Hariprakash, Harshitha, “IoT Based Home Automation using NODE MCU”, International Journal of Engineering Science and Computing, May 2018.

[19] Bröring, Arne & Charpenay, Victor & Anicic, Darko & Puech, Sebastien, “NOVA: a Knowledge Base for the Node-RED IoT Ecosystem”, 2019.

[20] Lekic, Milica & Gardasevic, Gordana, “IoT sensor integration to Node-RED platform”, 2018, 1-5.

[21] Deepak B. Andore, “AWS IOT Platform based Remote Monitoring by using Raspberry Pi”, International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), October 2017, ISSN 2278-2540.

[22] Sharmad Pasha, “Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis”, International Journal of New Technology and Research (IJNTR), June 2017, ISSN 2454-4116.

Authors

Sujatha Rajkumar is an Associate Professor in Department of Embedded Technology, IoT and Sensors Division, School of Electronics Engineering, at VIT University, Vellore Campus. She received her Ph.D. from Anna University, Chennai in the field of Information security. She has 21 years of teaching and research experience in reputed Institutions. Her research area includes Internet of Things, LoRa Networks, RF and Communication, Information security and Cloud computing. Her focus is on investigating how information security can be enhanced in IoT Big data using Blockchain. She received Speaker Award at UK Asia Summit,2019, Cambridge University, UK and IETE Best ISF award in the year 2010 and 2011. She is a lifetime member of IETE professional society and has published research articles in peer reviewed Academic International Journals and Conferences. Filed Patents for the innovative idea related to LoRa enabled smart Agribot.

Dhruv Haribhakti is a student in Vellore Institute of Technology, Vellore. He is currently pursuing Bachelor of Technology in Electronics and Communication Engineering with specialization in Internet of Thing (IoT) and Sensors. He is well versed in various technologies including Internet of Things, Cloud Computing, Logical Programming, Data Structures and Circuit Designing. He has previously worked on a research on how to detect the muscle disorder if any using EMG Signals. He has attended a number of workshops based on E-commerce, Ethical Hacking and Internet of Things. He has previously worked as an intern in Bombardier Transportation for 2 months.

Shreyas Loya is a student in Vellore Institute of Technology, Vellore. He is currently pursuing Electronics and Communication Engineering with specialization in Internet of Thing (IoT) and Sensors. He is well versed in various technologies including Internet of Things, Cloud Computing and Web Development. He has previously worked on a research paper involving Myoware sensors and muscle disorders. He has been awarded a certificate by the Maharashtra State Council for the Production of Biodiesel from coconut oil. He has previously worked as an intern in Reliance Jio Infocomm Ltd. He has also been certified by Indo-American Society for “Excellence in Public Speaking and Communication Skills”.

Dr Siva Kumar Subramaniam is a Senior Lecturer at Universiti Teknikal Malaysia Melaka, where he has been since 2006. He completed his Diploma in Electronics Engineering from Politeknik Ungku Omar in 2002. Siva Kumar Subramaniam than received a B.Eng in Electronics Engineering (Industrial Electronics) from Kolej Universiti Teknikal Kebangsaan Malaysia in 2006 and M.Sc in Electronics Engineering from the same institution in 2009 which is now known as Universiti Teknikal Malaysia Melaka. He received his PhD in Electrical Engineering and Electronic Research from Brunel University London, United Kingdom in 2017. His research interest is in wireless sensor network using IEEE 802.11 and IEEE 802.15.4 standards. He has a particular interest in a wireless sensor network, IoT system, system integration and consumer electronics system with a good track record of successful industrial collaboration for the past 10 years.