Pollution Based Intelligent Navigation System using Dijkstra's Algorithm

Dhruv Haribhakti, Vedant Chaubey and Shreyas Loya| Prof. Sujatha R.| SENSE

Motivation/ Introduction

Navigation has become an essential part of our lives today. Using Google Maps, we can check our live location as well as find out the minimum distance that we need to travel in order to reach our destination. Pollution, however, is on the rise and 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. One thing we can do to help is re-route the paths of the vehicles from highly polluted areas to areas having lesser pollution. Hence, we came up with this idea so that using the same navigation application, we can add the pollution detection and analyzing concept so that the user can calculate and assess the route having the least pollution.

SCOPE of the Project

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

Methodology

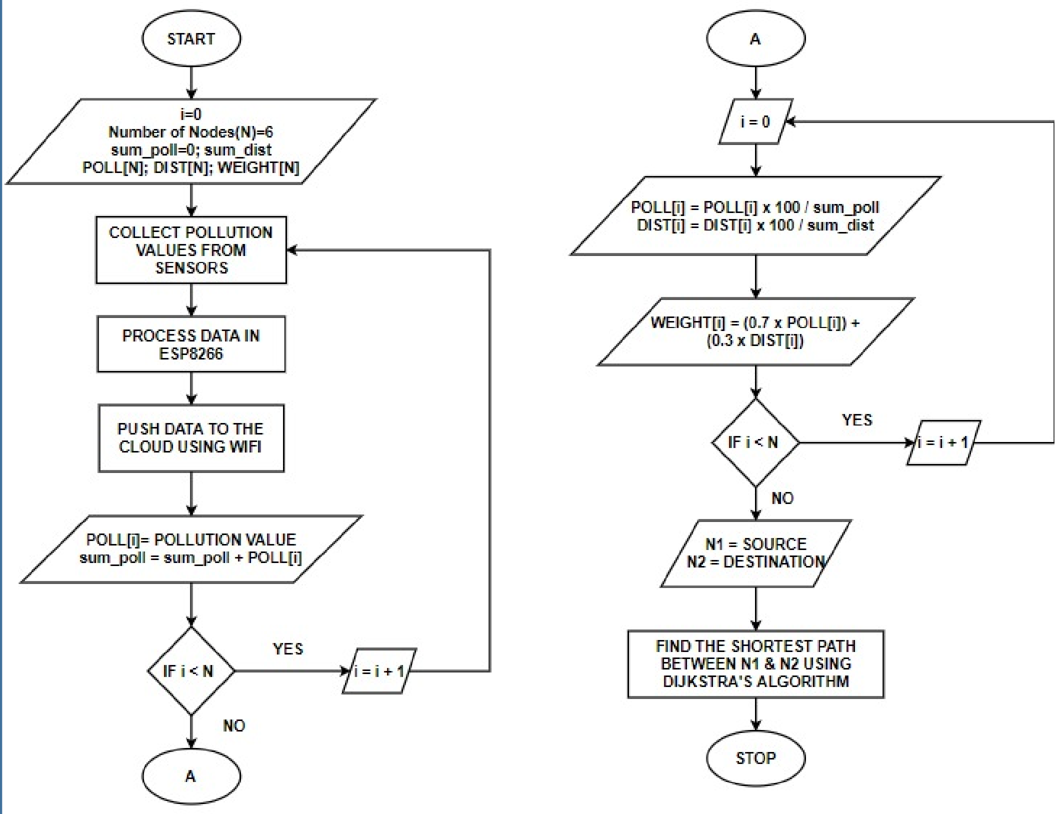
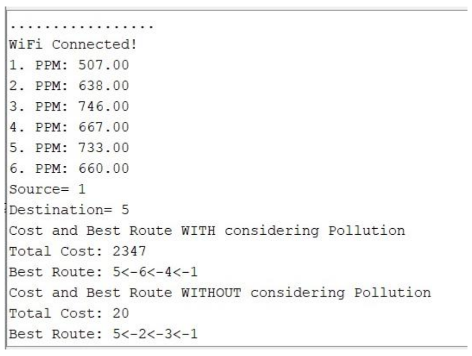


Figure 1. Process Flow

Pollution data needs to be collected from various sensor nodes(MQ-7 sensors) throughout the area. We define the number of iterations as 'i' and we assign variable 'N (here, 6)' to the number of locations in total. Now data is collected from the MQ-7 sensors and is pushed into a microcontroller. To do this, we make use of a 4051 MUX because there is only one analog pin available in the NodeMCU board which we are using. Hence the MUX acts as an intermediate and sends the pollution data one by one into the analog pin of the NodeMCU. Once we have the sensor data with us in the microcontroller, we consider the distance between the nodes. 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, we take the ppm value of the first sensor and divide it by the sum of the overall pollution ppm of all the N sensors combined (sum_poll). Once done, we multiply this value by 100 in order to get the percentage value. We carry out the same process for all the N sensors. Similarly, we calculate the percentages of the distances between the nodes. 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. We take the values as 70% since it is important to give more preference to the pollution and we keep 30% weightage for distance since it is important that we get the shortest route possible from the source to the destination. Now, the overall weights we 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, we get the optimum path as the output. Data is then pushed to the Firebase cloud using Wi-Fi. Data is received in the Real-time database which we have created in the cloud . Simultaneously, the application which we have built takes the data from this Real-time database and shows us the optimum path to take so that we encounter the least pollution as well as the least distance while travelling from the source node to the destination node.

Results

After implementing the code on the NodeMCU board, the sensors gave their output in the unit PPM (Parts Per Million) as shown in Figure 2. Once we had the pollution data and the distance between the nodes, we calculated the weights which were a combination of the pollution values and the distance values.



```
................
WiFi Connected!
1. PPM: 507.00
2. PPM: 638.00
3. PPM: 746.00
4. PPM: 667.00
5. PPM: 733.00
6. PPM: 660.00
Source= 1
Destination= 5
Cost and Best Route WITH considering Pollution
Total Cost: 2347
Best Route: 5<-6<-4<-1
Cost and Best Route WITHOUT considering Pollution
Total Cost: 20
Best Route: 5<-2<-3<-1
```

Figure 2. COM Port

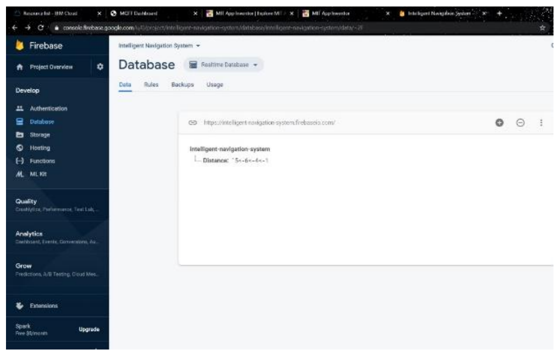


Figure 3. Firebase Real Time Database

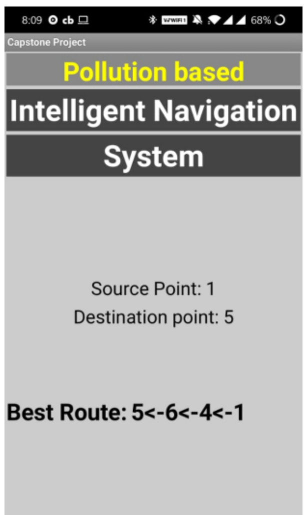


Figure 4. App Screen

Dijkstra's algorithm was calculated by taking the source node as 1 and the destination node as 5. The shortest route '1→4→6→5' was found out when pollution was considered, and the total cost was also shown (2347). When pollution was not considered, we got the route as '1→3→2→5' with the cost 20.

This data was then sent to the Google Firebase cloud. The best route after considering the pollution was displayed in the Realtime Database in the cloud as shown in Figure 3.
After the data has arrived in the cloud, it pushes this data (the route) into the application that we have built. As we can see in Figure 4, the optimum route is displayed in the mobile application.

Conclusion/ Summary

The system proposed by us displays the paths having a combination of the least pollution and the least distance between the nodes (giving higher priority i.e. 70% to the pollution and 30% priority to distance) as well as the path having the least distance between the nodes completely disregarding the pollution data. This data is showcased on the cloud as well as on a mobile application so that it can be accessed easily with use anywhere in the world. Using this data, the user will be able to travel safely avoiding the areas having high amounts of air pollution and low air quality. The data can be fed into the navigation applications of today in real time and can be used to direct the user so that he can take the least polluted optimum path to reach his destination.

Contact Details

dhruvharibhakti7@gmail.com, shreyasloya1@gmail.com, vedantchaubey@gmail.com

Acknowledgments/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. 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.
3. Priyanka Phadtare, Shemant Kumar Singh, "Intelligent Navigation System using Air Quality Index", International Research Journal of Engineering and Technology , 2018.
4. 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.
5. 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.
6. 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.
7. 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.
8. Amr Elmasry and Ahmed Shokry, "A new algorithm for the shortest-path problem", Networks, Vol.-74, 21 December 2018.