International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I3P244

Volume 71 Issue 3, 423-433, March 2023

Original Article

Smart Traffic: Traffic Congestion Reduction by Shortest Route * Search Algorithm

A. Lakshna[1], S. Gokila[2], K. Ramesh[3], R. Surendiran[4]

[1,2]*Department of computer applications, Hindustan Institute of Technology and Science, Chennai, India.*
[3]*Department of Computer Science Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India.*
[4]*School of Information Science, Annai College of Arts and Science, Kumbakonam, India.*

[1]*Corresponding Author : lakshnaa73@gmail.com*

Received: 10 October 2022 Revised: 07 March 2023 Accepted: 24 March 2023 Published: 30 March 2023

Abstract *- Traffic congestion is a major issue on the city side, resulting in an increased level of traffic and negative impacts on the environment and public health. In this research, the shortest route * search algorithm proposed a solution to find the simplest route with the minimum duration in traffic congestion. The shortest route * search algorithm searches for the simplest route from the origin to the destination while considering current traffic conditions. The shortest route * search approach involves traffic data pre-processing of previous records and current traffic data to create a weighted graph representation from the road network. The proposed algorithm works only on the best route and concentrates on the nearest shortest node to determine the simplest path. This approach optimizes the time complexity by avoiding search time for all nodes. The algorithm was evaluated on a traffic dataset, which resulted in 97% of high accuracy. The shortest route * search algorithm was compared with the existing algorithm. The results showed that the shortest route * search approach significantly reduced the travel time in traffic conditions, demonstrating the potential of a shortest route * search algorithm to improve transportation efficiency and reduce environmental impact.*

Keywords *- Shortest route prediction, Shortest route * search algorithm, Shortest paths, Fastest route, Minimum travelling time.*

1. Introduction

Smart road traffic congestion reduction refers to using new technology and data analysis to alleviate road traffic congestion. In a smart city, the traveller cannot reach the destination on time because of heavy traffic congestion. By reducing traffic congestion, graph theory algorithms are used to reduce the level of traffic by identifying the road structure; if any damage or construction work is going on, it will observe and ask the traveller for comments. According to them, the result will be displayed to an upcoming traveller. Suppose a road blockage implies that the upcoming vehicle is redirected into the shortest alternate path. Here, the shortest path represents the simplest way to the traveller, but it would not redirect them into a critical shortest route, which would be a smooth flow to the traveller. Thus, reducing the congestion level helps other travellers without struggling with traffic congestion. Based on previous results, the traveller can avoid traffic congestion. If the traffic condition is displayed to the upcoming traveller, it is easy to travel because they can change the plan accordingly (change the route). This method helps passengers to travel more easily. Sensors were placed on the roadside to collect real-time traffic data, and preliminary data were used to predict the traffic level. Finding the shortest route to reach the destination can be achieved through various measures such as intelligent transportation systems, dynamic traffic management, real-time traffic monitoring, and route navigation algorithms. Many travellers are unable to travel on the correct path, sometimes the map shows an incorrect route, or it would not show the correct path, whether it needs to take the bridge route in an upward direction or else they need to take a downward direction under the bridge road (subway bridge) to prevent incorrect route identification; here, it is used to detect and identify the correct path.

The correct road is indicated in yellow. Many travellers are stuck in traffic but do not know the correct path. Here, describes the reason for the traffic to the traveller. Path identification helps the traveller reach the destination without wasting time by taking the wrong long route. Currently, most travellers do not know the shortest route to reach their destination. Therefore, maps were used for navigation. Sometimes maps lead to improper routes that cause the traveller to ride on very long routes; sometimes, travelling on the longest road at night is unsafe. Sometimes a traveller is struck by a blockage in the road, thus leading to heavy traffic; therefore, they cannot reach their destination on time. Here, it will be uploaded to the traveller who passed the same lane and given feedback such as construction work, accident, metro work, subway, underwater pipe connection, and fiber connection list will be displayed to the traveller so that they can select the reason for congestion.

Sometimes traffic may vary from the urban road and highway road if the road width is small and many vehicles are waiting, resulting in traffic. In particular, curve roads or cross paths in highways sometimes record many congestions and accidents. High-speed vehicles like Lamborghini, duke cause accidents leading to traffic. New drivers recently learned driver using vehicles without L board, rash driving by teenagers, and drunk drivers. Due to natural calamities, people are stuck in traffic due to heavy rain, and also, in peak hours, everyone struggles in traffic. There are five main attributes to predict road congestion they are the type of road, vehicle characteristics, driver behaviour, time, and weather. The type of road will identify the road structure and what kind of road it is. Suppose traffic is found, they will divert into the shortest alternate path. The first step here is identifying all possible routes. The second parameter is to identify the vehicle characteristics according to car model, speed, and acceleration can be identified. Only emergency vehicles can drive fast. The sports vehicle is driving very faster; that type of rider can also be identified. The third parameter, driver behaviour, should be noted because there are many types of drivers causing traffic. The beginner driver does not know how to drive smoothly. The drunk and drive and rash driving drivers. The fourth parameter is time; they should postpone or pre-pond the travelling according to peak hours. It priorly informs the traffic level according to that plan for your trip. The fifth parameter is whether natural calamities also cause traffic, so everyone should monitor the weather condition before travelling.

Various algorithms, such as genetic algorithms representing crossover and mutation, Swarm Particle Optimization, and Artificial Neural algorithms, can be used to predict the shortest path. First, they used a time-space model to predict traffic. Then the ride-sharing algorithm was used for the communication purposes of intelligent recognition, location, tracking, monitoring, and management. Swarm Particle Optimization represents using fish schooling to optimize the search path. Bee colony optimization has been used to solve ride-sharing problems [1]. By analyzing traffic data and guiding the road to improve the urban area using graph theory, it constructs self-organizing feature map neural networks [2]. The shortest route * search algorithm will search for the shortest route to reach a destination, the suggested route will be without traffic, and the road structure is also fine; only the smoothest and simplest route will be suggested to the traveller to make travelling easier. Further, this research is classified into six sections: Introduction in Section I, Contemporary review in Section II, Proposed method in Section III, the proposed outcome in Section IV, result and discussion in section V, and followed by conclusion in section VI.

2. Literature Review

The System Optimum introduces path control for better transportation, and the minimum control ratio is mathematically and numerically applied to a real-world dataset. Finding the path with the minimum control ratio is a mathematically and numerically worked, unreal-world dataset. Finding the path with the minimum ratio has been concluded with a result of 96%, but it is only for autonomous driving vehicles [3]. Using tabu search to solve the difficulty in pathfinding by using the swap procedure, the traveller can switch paths where less traffic is found compared to other routes [5]. Traffic is dependent not only on heavy vehicles but also on the total distance. The result is 90% in achieving the shortest route to the traveller using aerial images. The aerial images were captured by an unmanned aerial vehicle using a drone camera. By ignoring the rush hours, they need to update the cost function co-efficiently so that they can reach the destination in the shortest time [6]. To minimize traffic congestion, multipath leads to achieving the result. The pheromone model increases the system's efficiency to reduce computational issues; however, the result of reducing the error rate is only 20% [7]. Old methods, such as the Dijkstra algorithm, cannot achieve high performance. Predict superfast shortest routes with the maximum number of nodes. This results in low efficiency [9]. Predicting and analyzing a simple path reduces the data pre-processing time and space costs [8, 10].

Analyzing traffic using global positioning system data and vehicle detectors. The reason for the variation in traffic data is that the first accidents cause major traffic, and the second bad weather, such as rainy days, may cause heavy congestion. The traffic level N-curve pattern from a convolutional neural network was reduced [11]. It detects the density pattern, which is the same as detecting the traffic pattern in a congested area. By using speckle warner detection, it detects the pattern very easily [13,16]. Traffic congestion is a major issue that addresses this challenge with many routes from convolutional networks to identify the congestion level based on the traffic speed in a particular network. The experimental results showed an 18.9% prediction error [14]. A deep neural network was used to predict the minimum travelling time through connected vehicles using a root mean square error of 0.255. Traffic flow was predicted using past traffic flow data [15]. Find the shortest path using neural-network-driven prediction to identify the new shortest path with high speed in producing the result for the traveller.

It is executed in a highly dynamic environment [17]. In transportation networks, a graph wavelet is used to identify the region in the form of a graph structure. The combination of neural networks provides more information on traffic with better results [18]. Real-time and space complexity analysis using a real-world road network for the (P) result of the path, with a target (t) and alternate path (K) to find the space for travelling in the shortest path [4,19]. Many people struggle in traffic to reach their destination, but owing to heavy congestion, they cannot reach their destination on time [20, 21]. The traveller cannot easily view the result and avoid the correct lane because of heavy traffic [22]. It does not make travel smooth without any hectic situations. It will lead them to travel with stress, so they cannot reach their destination on time [23].

2.1. Heap-based/Enhanced Bellman Ford Algorithm

This algorithm has a major issue: time complexity. It operates very slowly on larger graphs with more edges. It is inefficient for dense graphs and probably does not lead the traveller to the shortest path. It identifies the negative weight cycle; therefore, it cannot determine the shortest path. This is because the distance value for the node on the cycle will be updated indefinitely, which makes it impossible to determine the shortest path. Multiple iterations are required to search for the shortest path for a larger graph, which consumes more time to predict the result. Only retrospective traffic data were used to analyze and identify the shortest route for drivers. This suggests a route to reach the destination [25]. However, the shortest route search algorithm uses different types of input data to improve the accuracy of traffic results. A mathematical model is used to predict traffic and reduce congestion levels. The two methods are useful for identifying the vehicle type and minimizing the cost function; only 5% is better than in the previous work. For better performance, the percentage level must be improved [26].

The indexing algorithm is efficient and performs a guaranteed query-processing algorithm. Based on the experimental results of the algorithm, it can speed up query processing time by two orders of magnitude without consuming more index space than the existing approach [27]. Finding the simplest route through multiple query subsets. If the road is zigzag, it is considered a 1N sharing computation. The result is enhanced from one region to another with a bounded error [29]. Find the shortest path using multiple graph-matching methods in the online and offline modes and combine all shortest paths into a single-source shortcut path. It is efficient to mix all routes and find the shortest route among them [28, 30].

2.2. Enhanced Dijkstra Algorithm

The enhanced Dijkstra algorithm cannot be used in graphs that contain negative-weight edges because it assumes that all edge weights are non-negative. Thus, it does not produce the correct result. It requires more time to work on large graphs. Here, the greedy approach was used to select the edge with the smallest weight for each step. This does not produce a globally optimal result. Running for every node as the source is necessary, which is more time-consuming for larger graph models. All the suggested paths are analyzed using two nodes from the graph network, and the best route is determined. The result shows the best path, but it takes a large amount of time to predict the result because it finds all paths [31].

The shortest route * search algorithm does not waste time identifying the shortest path; here, it finds the shortest path without wasting time by searching all the other nodes. Using the Dijkstra algorithm, the dynamic shortest path was calculated by updating the list of affected vertices according to the updated shortest path [32, 33]. Chronicle data are used here to better predict traffic. First, the model was trained on various sources for better initialization. This method is a combination of spatial features. Experiments with this method obtained better results [34].

2.3. Time-dependent a potentials algorithm*

The time-dependent A* potential algorithm fully depends on the heuristic function; sometimes, it selects the worst node, which may lead to an inefficient search process. This function is not admissible because it does not produce the correct results; therefore, it may find a suboptimal path due to this issue. The Time-dependent A potential algorithm works very slowly in highly connected graphs [36]. However, the shortest route * search is very fast for predicting a better result. The conventional multipath algorithm uses a goal direction search with a minimum number of predicted nodes; however, it is an amateurish spatiotemporal neural network. This differentiates all routes and determines the best route.

The breadth-first search was used to identify many paths; however, only repeated paths were found. In a depth-first search, the solidity of the route decreases. The main drawback is that the goal of finding the best route is not a prime road or an excellent path [37]. Time-dependent vehicle routing problems detect the speed of a vehicle, travel time, waiting time, service time, time window, and vehicle carbon emissions.

Ant colony optimization is used here to avoid traffic congestion and reduce costs [38]. Minimizing the routing cost introduced a model to reduce traffic congestion and the issue of detecting the speed of each route. Minimizing this multipath traffic cover results in producing minimizes the path cost. The result shows that 25% cost is decreased to analyze alternatives; sometimes, it may vary due to driver behaviour for positive and negative road angles [39].

This study aims to improve the accuracy of the heuristic shortest-path algorithm. Three existing algorithms were compared with the proposed algorithm: the heap-based/enhanced Bellman-Ford algorithm enhanced Dijkstra algorithm and the time-dependent a* potentials algorithm discussed in the background study. Shortest route * search algorithm. The aim of this study is to improve the accuracy of the heuristic shortest-path algorithm. The shortest route * search algorithm works very well in finding the shortest path by using additional five parameters the type of road vehicle characteristics, Driver behaviour, time, and weather (r, v, d, s, w). this plays a major role in shortest route prediction because of additional inputs.

Implemented in jupyter notebook using python programming language, it resulted in high accuracy of 97% in prediction and reduced the error rate from 30 minutes to 9 minutes and 30 seconds. It is compared with the enhanced bellman ford algorithm, Dijkstra, and time-dependent A* potentials are very less in accuracy and latency, but the shortest route * search algorithm works very well in both situations. Hence the shortest route prediction algorithm helps the traveller to find the shortest alternate path without getting stuck in traffic so that they can pre-plan their travelling time.

3. Shortest Path Selection using the Shortest Route*Search Algorithm

Using a heuristic approach, the navigation cost is predicted from any point to a destination point. From the starting node, it predicts the route with the lowest estimated cost until it reaches its destination. If it finds a destination nearby, it will not search for any other node. Connected nodes are used to connect neighbouring nodes, forming a continuous network of edges. The Traffic star search algorithm was used to reduce the travel time and distance of the travelling route. The edges store a hint that lists the shortest paths to the nodes to which they lead. A heuristic function is a mathematical function that provides cost estimation and is used in optimization, decision-making, and pathfinding problems. A heuristic function determines the remaining distance from the starting state to the ending state. It estimates the path and gives priority to the nearest path. Here, the additional parameters r, v, d, s, w. represents the road vehicle characteristics, driver behaviour, time, and weather, respectively. This helps to reach the destination in the simplest path. The simplest path refers to the absence of any traffic or delays that can reach the destination within the minimum duration Detecting traffic congestion levels is analyzed through sensors such as infrared, laser scanners, and radio frequency identification tags, which are used to find the traffic congestion level in binary format (0 and 1). Improving further by including a fine path/correct route. If the route is good, it needs to be checked; otherwise, if any blockages are found, it redirects it in the simplest way, not the critical shortest route.

The drawback in traffic is road blockage, which leads to a long route (i.e., it sometimes shows the path on an interior roadside), and it is better to avoid those roads and take a national highway, as it leads to a straight road without taking any diversion. Overcoming this problem suggests the simplest way to reach the destination. Owing to less accuracy in the results, the performance of vehicles overlaps, and crashes occur. The best solution to avoid these problems is to provide better results with minimum travelling time. The shortest route * search algorithm was used to predict a better result with 97% of the maximum accuracy result to solve this problem. It provides multiple paths to users so that they can select the route they need to travel.

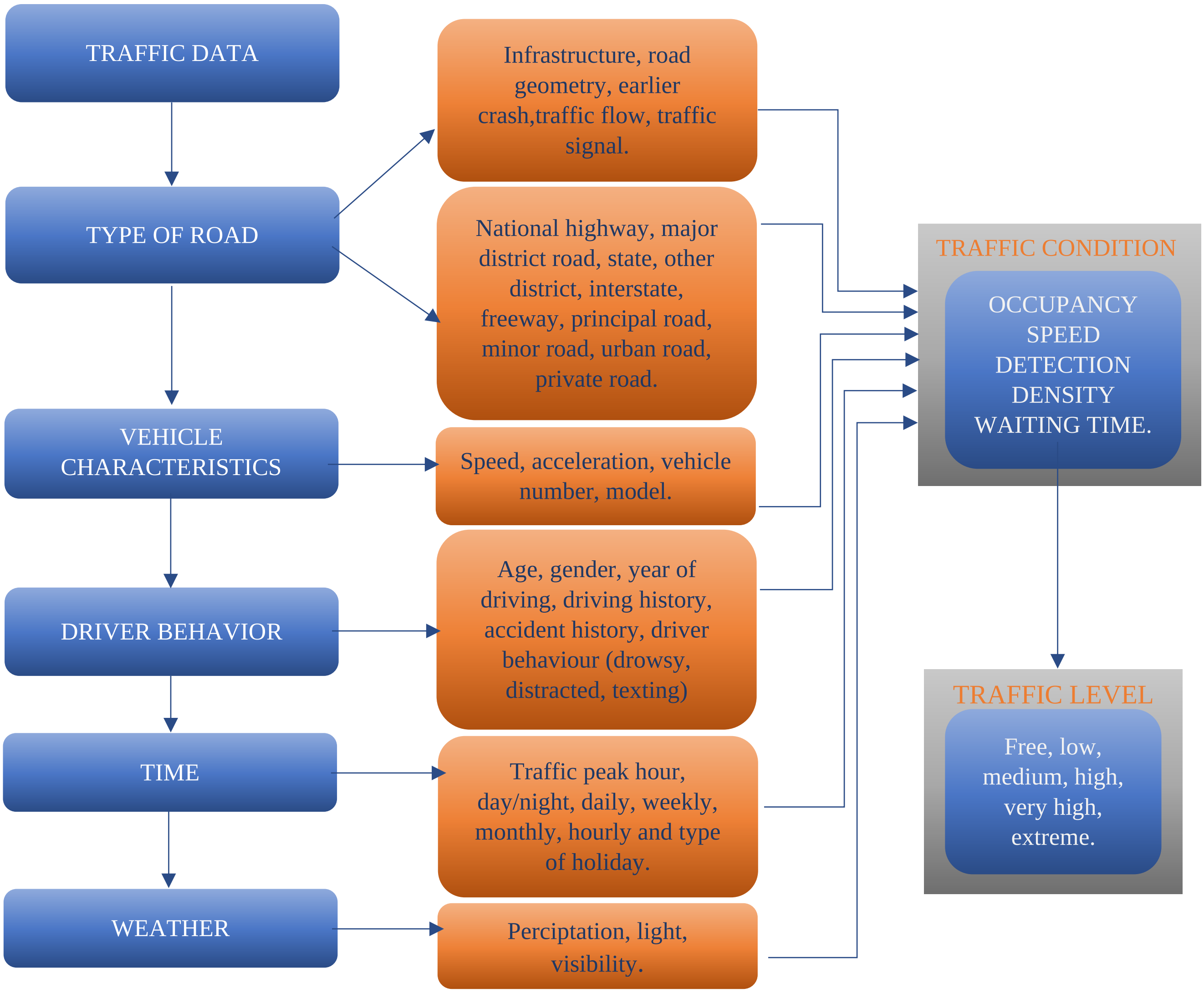

Fig. 1 General view of traffic data

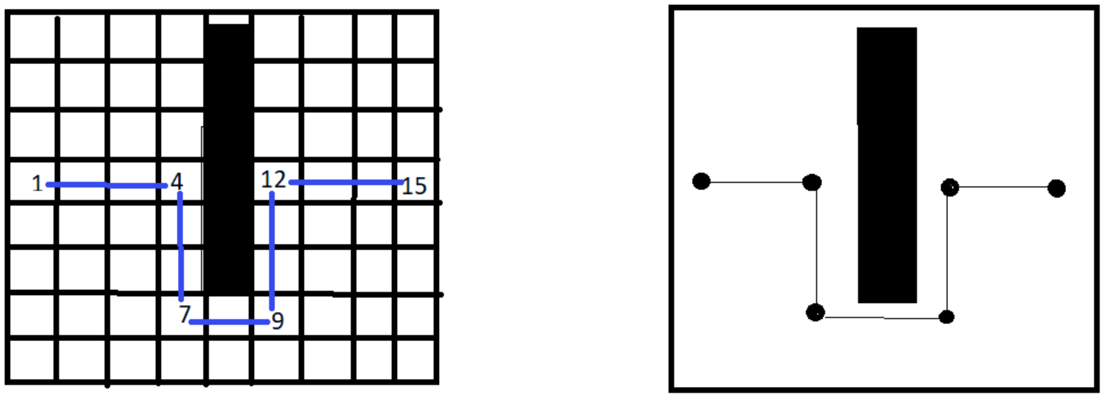

Fig. 2 Illustration of the Shortest route * search algorithm searching shortest path from the 1st grid to the 15th grid.

The reason for developing an algorithm is to search for the simplest route to arrive at the end node. Here, searching for a better algorithm to reach the destination. Here, finding the best algorithm to reach the goal using the heuristic method at the reaching point is very easy. Different types of scenarios exist in the graph theory used to reach the destination. The old method searches for all paths to reach the destination. According to existing work, searching all paths is a waste of time. The aim of this study is to improve the accuracy of the heuristic shortest-path algorithm.

4. Proposed Outcome

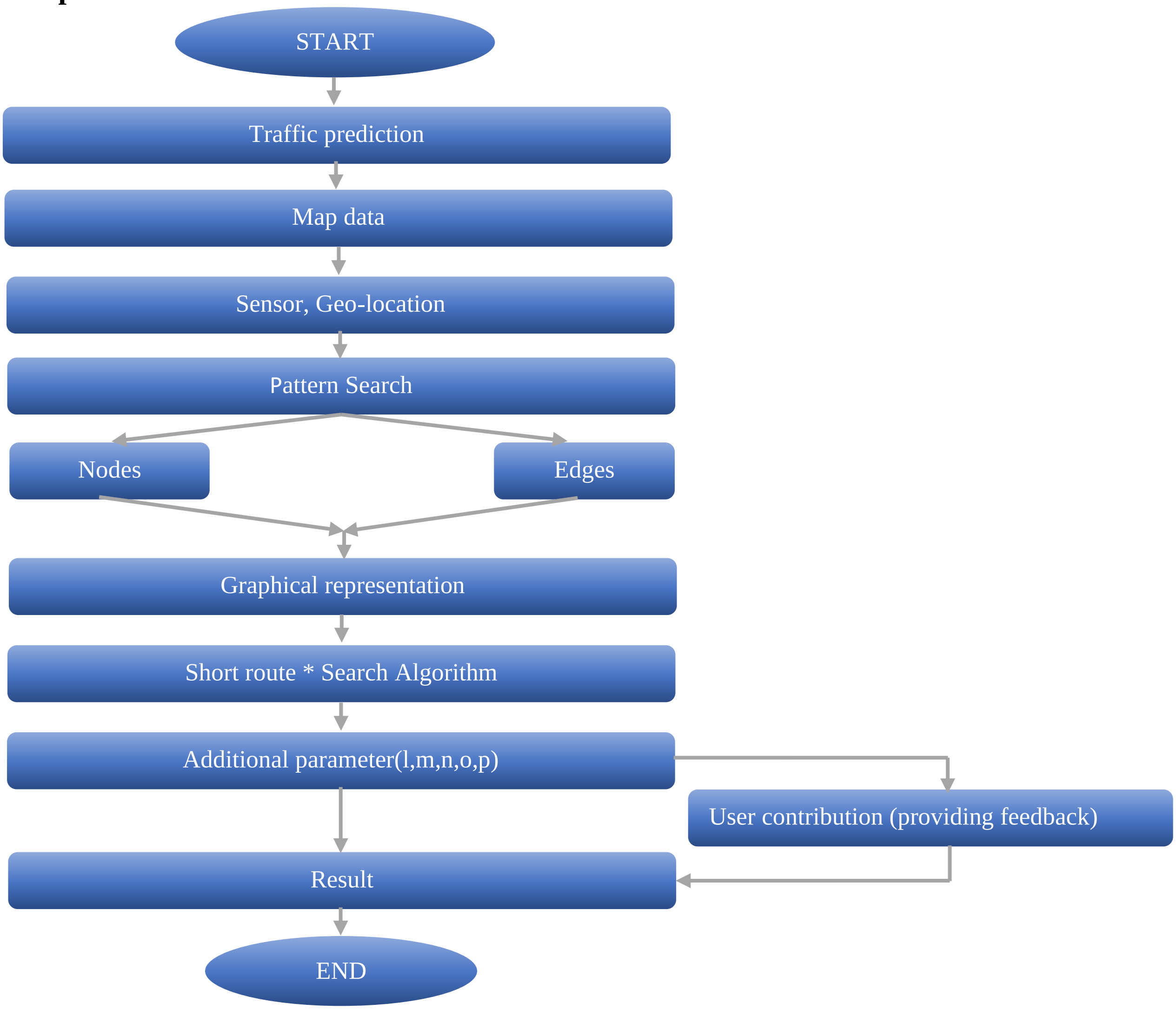

Fig. 3 Sequence diagram for shortest path prediction

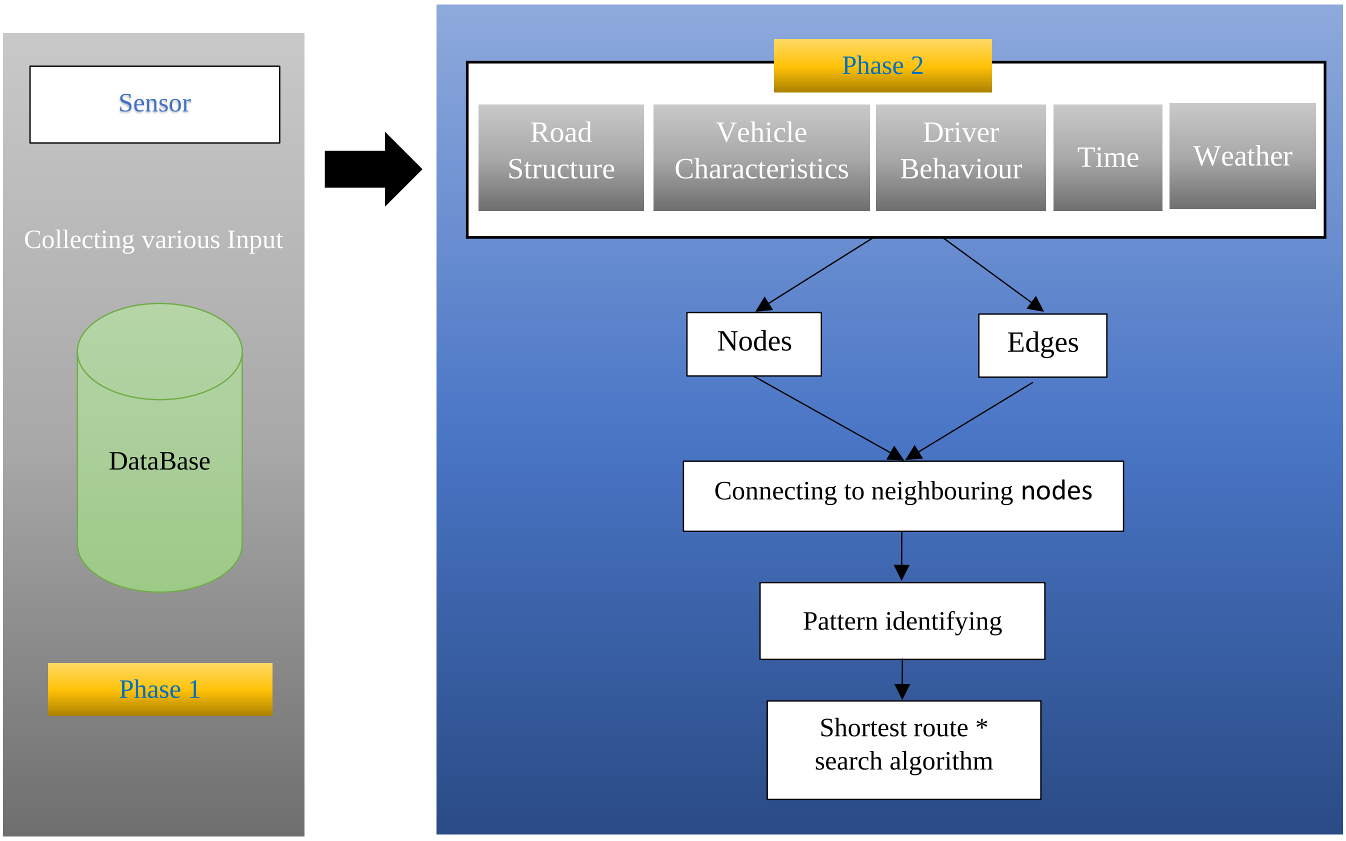

Fig. 4 Architecture diagram for smart traffic

Accurate traffic prediction using current and factual data with additional parameters l, m, n, o, and p. They are road type, traffic flow level, traffic condition, type of vehicle, weather, and time. First, the road types were categorized into major roads, motorways, national highways, trunk roads, principal roads, minor roads, urban roads, rural roads, private roads, freeways, no lanes, interstate roads, and major district roads. The second is road attributes, such as density, occupancy, waiting time, and speed, and the congestion level is represented by free flow, low, medium, high, very high, and extreme. Vehicles are represented by vehicle characteristics, speed, and acceleration, and the weather is identified by perception, light, and visibility. Time was represented as day/night, peak hours, weekly, monthly, daily, and hourly. Driver information is based on driver behaviour, such as drowsiness, texting, distraction, accident history, year of driving, gender, and age.

The smart traffic architecture explains the working process. The first step is gathering the different types of inputs through a sensor. In previous work, everyone predicts the traffic according to the number of vehicles. Here, the shortest route star search algorithm identifies various attributes that help predict a very accurate traffic result. The attributes are represented as w-weather, s-time, d-driver behaviour, r-road structure, and v-vehicle characteristics. After getting these numerical data, it is pre-processed into a finite structure where it removes unwanted and duplicated data. It runs under jupyter notebook in a python programming language. Where 2,56,61,847 dataset is used after data-preprocessing, and it came into 42,54,811. The algorithm time complexity is also noticeable because it reduced 30 minutes into 9.30 seconds. Using the shortest route * search algorithm reduces the error rate. It produces the result within a short duration so the traveller can easily avoid congestion and take the shortest alternate route to reach the destination.

Here, concentrate only on the selected path, the starting and ending nodes. The line is connected between two nodes called edges. It works on edges because it searches for a simple route without wasting time by searching for all irrelevant routes. Adding parameters r, v, d, s, w. in the shortest route * search algorithm will result in high efficiency because here include all types of information related to traffic congestion. It identified the major cause of traffic. First, identify the type of road through previous data. It identifies the daily average traffic flow. Second, it observes the vehicle's characteristics, like speed, acceleration, and type of vehicle. The third step is identifying driver behaviour, which causes a major impact on traffic and accident. Another main important node is timing. The peak hours are observed hourly (day/night). Due to bad weather conditions on rainy days, congestion levels will be high. The Shortest route * search algorithm used this parameter and highly accurately predicted the traffic level. If congestion is found, it diverts the traveller to the shortest path without traffic, thus resulting in smooth vehicle flow. This is the only proven solution for predicting traffic with high accuracy.

T	- Total cost of the node
D	- Distance between two nodes
H	- Heuristic estimated distance from starting point to ending point.
R	- Type of road
V	- Vehicle characteristics
D	- Driver behaviour
S	- Time
W	- Weather
Goal	- Ending node
First node	- Parent node

Number of iteration process i and j

$$X=Fnode.x+i \quad (1)$$

$$Y=Fnode.y+j \quad (2)$$

X=width
Y=length

Searching the nearest neighbour node X & Y

$$Distance = F.X\_N.X+F.Y\_N.Y \quad (3)$$

Tentative score>=N.Dscore

$$Current\ node.T=N.Dscore+tentativescore \quad (4)$$

$$Current\ node.T=N.Hscore+sum\ of\ traffic\ parameter \quad (5)$$

$$Current\ node.T = current\ node.D + current\ node.H+ r+v+d+s+w. \quad (6)$$

The shortest route * search algorithm prioritizes the shortest node to reach the end part. Pseudocode for Shortest route * search algorithm is given that D, H score is initialized with an additional parameter r, v, d, s, w. Initialize the starting node to the ending node, then set the final node to a goal which is the destination point. Add the parent node, represented as the first node, then remove the starting node after a number of iterations distance is calculated from the starting node to the neighbour.

```
    step 1    Initialise DScore
    step 2    Initialise HScore + r, v, d, s, w.
    step 3    T= D+H+r, v, d, s, w.
    step 4    Initialise SNode and Enode
while loop
    step 5    if SNode > 0
    step 6    if <SNode score
    step 7    Initialise FNode from the grid
    for each loop
    step 8    if node.DScore < Fnode.DScore
    step 9    if Fnode = goal
while loop
    step 10   Check Fnode is ≠0
    step 11   Initialise path
    step 12   parent node - Fnode
    step 13   Add Fnode
    step 14   Remove node from SNode
    step 15   Add node to Enode
for loop
    step 16   i -1
    step 17   i < or =1
    step 18   i +1
for loop
    step 19   j-1
    step 20   j is <or =1
    step 21   j+1
    step 22   if i = 0 and j = 0
Continue to loop
    step 23   X =FNode.x + i
    step 24   Y = FNode.y + j
    step 25   if X < 0
    step 26   X >= width
    step 27   Y < 0
    step 28   Y >= length
Continue to loop
    step 29   if X, Y = 1
    step 30   Continue to loop
    step 31   Initialise new node = neighbour
    step 32   neighbour = {neighborX and neighborY}
    step 33   if Enode (neighbour)
Continue to loop
    step 34   Initialise distance
    step 35   distance = Fnode.x-neighbour.x + FNode.y-neighbour.y
    step 36   tentativeScore = FNode.DScore+distance
    step 37   If distance = SNode + neighbour
    step 38   else if tentativeScore >= neighbour.DScore
Continue to loop
    step 39   T = {neighbour.parent + Fnode}
    step 40   T ={neighbour.DScore + tentativeScore}
    step 41   T =={neighbour.HScore + sum of traffic parameter}
```

5. Result and Discussion

The pre-processing of the traffic data is done by using these steps. Data cleaning and reduction are used to remove the unwanted data and split the data according to the need, then connect a graph through edges with neighbouring nodes. Graph optimization predicts the result using the shortest route * search algorithm with 97% accuracy. The original dataset's total amount is 2,56,61,847 after data pre-processing, such as data cleaning and data reduction. Thus obtained dataset amount is 42,54,811. Then, the data is partitioned into a structured format, like a type of road vehicle characteristics, Driver behaviour, time, and weather (r, v, d, s, w). By using this parameter, it is useful to achieve the result in a better way. Implementing the dataset into a jupyter notebook working under python programming language. The shortest route * search algorithm is implemented with obtained dataset and found a result with a 97% of accuracy level in prediction. Results in 92.10%, time-dependent potentials algorithm [34] result in 94.30% are predicted with low accuracy. So, compared with other algorithms, the shortest route* search algorithm works better with high accuracy because of the additional parameters r, v, d, s, w.

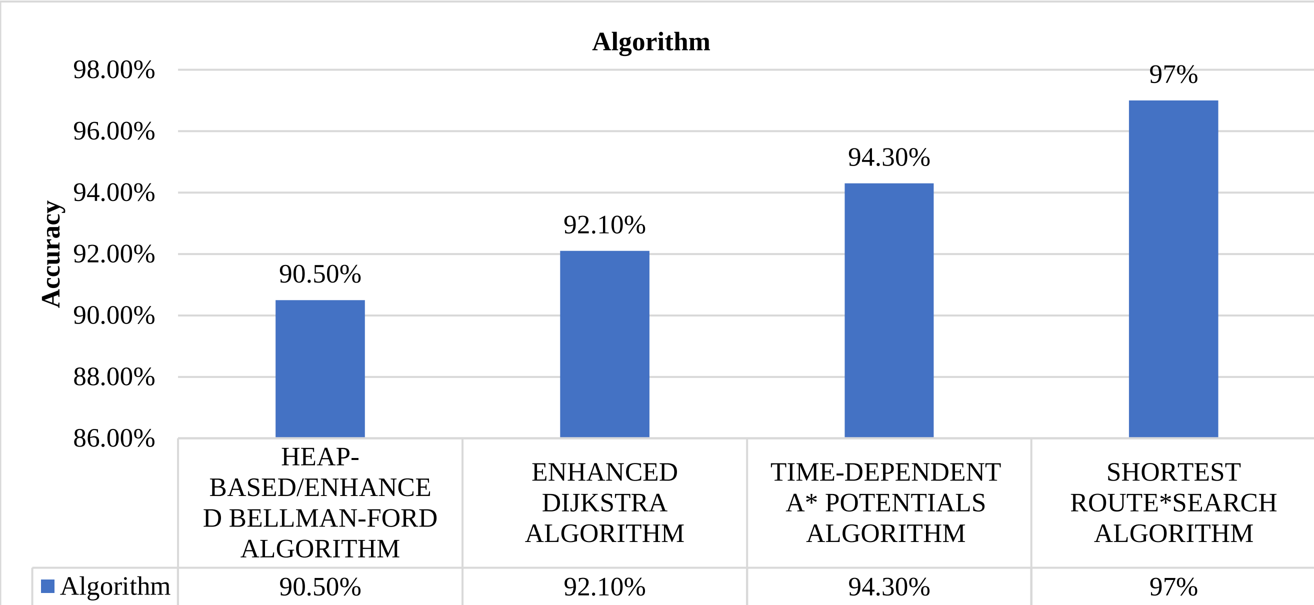

Fig. 5 Comparison between different algorithms

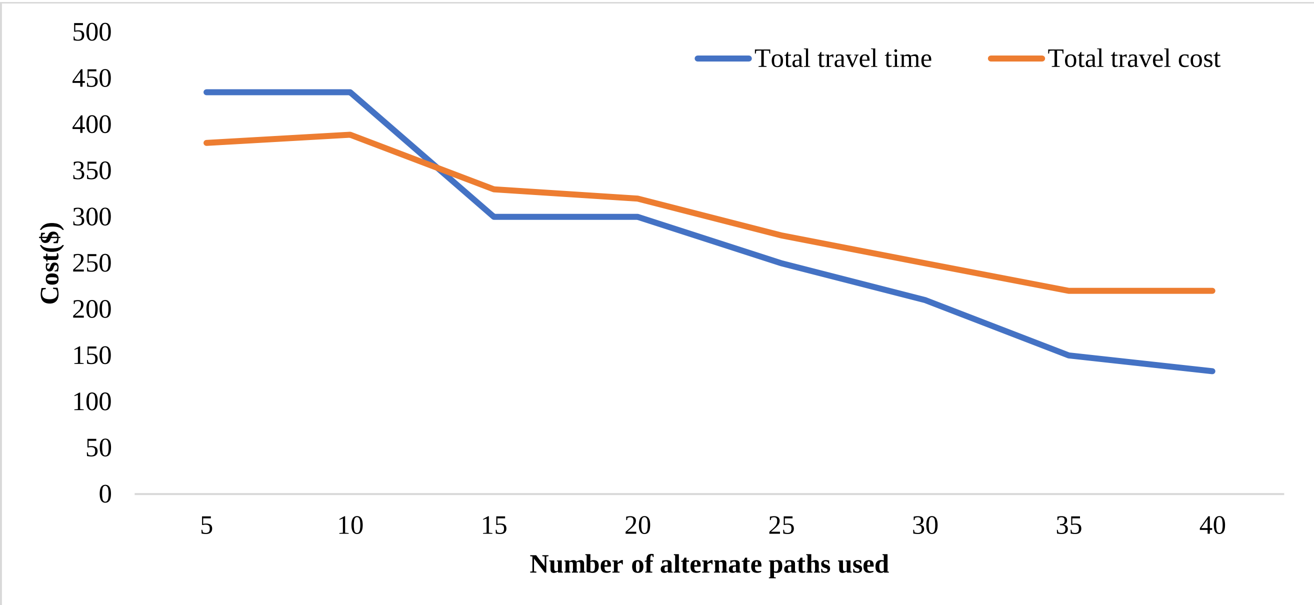

Fig. 6 Difference between total cost and total travel time

The difference between the percentage level of the ensemble algorithm. The existing algorithm heap-based bellman ford algorithm results in 90.50%, the enhanced Dijkstra algorithm results in 92.10%, time-dependent potentials result in 94.30%, and the shortest route * search algorithm results in high accuracy with 97%.

The big O notation is used to analyze and compare the efficiency of algorithms for solving various problems. Here comparing four types of an algorithm for time complexity. (O)n- a maximum number of operations the algorithm requires as a function of input size. It refers to the running time that grows linearly with the input size. If the size doubles running time also doubles. Compare the proposed algorithm with the existing algorithm and prove that the shortest route * search algorithm works better for predicting the shortest route to reach the destination.

The graph explains the difference between the result of different algorithms. The other algorithm result of heap-based/enhanced bellman ford algorithm - 90.50%, enhanced Dijkstra algorithm - 92.10%, and time-dependent a* potentials algorithm – 94.30%. The Shortest route * search algorithm results in 97% with the best accuracy when compared to other algorithms. Hence Shortest route*Search provides the best shortest route to the traveller. The aim of this study is to improve the accuracy of the heuristic shortest-path algorithm. There are many processes to identify the traffic, such as a global positioning system that can provide vehicle direction and location. Traffic cameras capture the real-time traffic flow image. Road network maps provide layout and intersection. Here sensors are used to collect various traffic information of traffic flow, volume, speed, number of vehicles, etc.

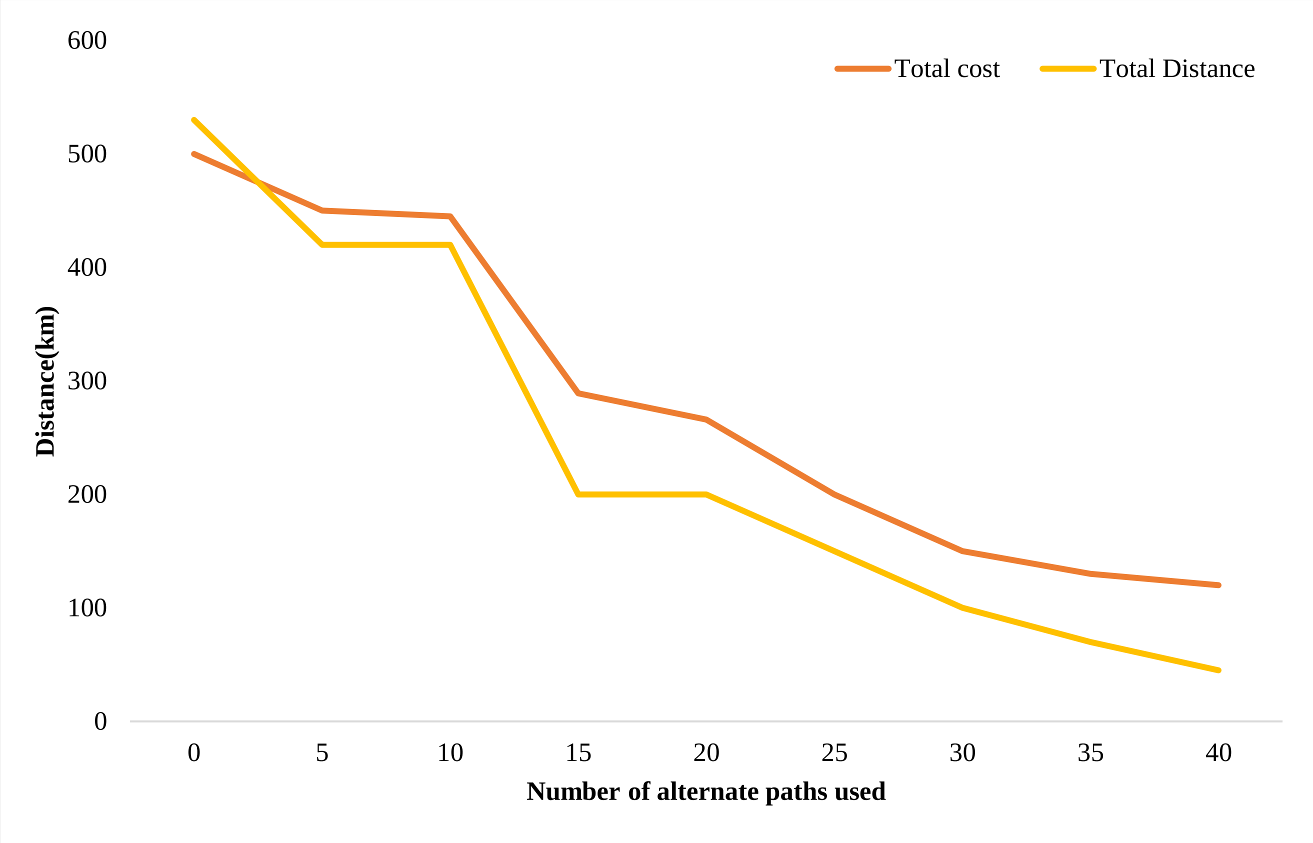

Fig. 7 Difference between total cost and total distance

The graph explains the difference between the actual path and the shortest path, thus reducing the total distance and travel time, which helps the traveller to reach the destination on time. Using the Shortest route * search algorithm makes it easy to find the simplest route, thus reducing the total cost. Here the comparison between total cost, travel time, and total distance for a number of alternate paths is used. The shortest route * search algorithm works in different stages. The first step is collecting the input data through sensors, which collect the number of vehicles passing through the area, road structure, climatic conditions, vehicle characteristics, road structure, and the historical record of driver behaviour also noted. These five parameters are a major source for identifying the correct output. Everyone used to take only the number of vehicle counts from the camera, but it takes more time to predict the congestion level here, which is the shortest method to find the congestion level. After identifying the traffic level, the traveller is allowed to take a diversion or take an alternate shortest route.

Using the shortest route * search algorithm, find the nearest neighbour node from the edges. The patterns are identified, and able to find the nearest shortest route to reach the destination is. This helps to reduce the traffic level, and they can reach their destination on time without getting stuck in traffic. After twenty-five iterations, the algorithm time complexity results in a reduced level from starting time to the ending time. In previous work, the time taken to predict the result is thirty minutes, but it takes nine and thirty seconds to complete the prediction here.

6. Conclusion

The Shortest route*search is an effective method to find the shortest path in heavy traffic congestion areas by pre-processing the previous and current traffic data with various input types and identifying the shortest path through graph theory. The starting node concentrates on finding the nearest targeted goal and then processes each step. The main advantage of the shortest route * search algorithm is that it does not waste time on spending any other unwanted nodes. Compared with other algorithms such as Time-dependent A*potentials, Enhanced Dijkstra algorithm, and Heap-based/Enhanced bellman ford algorithm, result in less accuracy displayed in graphical representation. Shortest route*search makes the fastest prediction of the shortest route with 97% of high accuracy. The aim of this study is to improve the accuracy of the heuristic shortest-path algorithm. Shortest route*search results demonstrate that the approach significantly reduces travel time in congested traffic conditions compared to existing routing methods. This could significantly minimize the environmental impact of travelling. While the proposed research work focused on a specific urban area, the shortest route*search approach can be applied to other regions and provide similar benefits.

Additionally, here acknowledge that there are limitations in the approach, including the need for real-time traffic data. However, this research provides a strong foundation for future work in this area. These findings will lead to further advancements in transportation efficiency and sustainability.

References

[1] Verma, K., and Paike, V, “Smart City Intelligent System Traffic Congestion Optimization Using the Internet of Thing,” *Arxiv Preprint Arxiv:1911.01286*, [CrossRef] [Google Scholar] [Publisher Link]

[2] Xiaowei Chen, Harry Haoxiang Wang, and Bin Tian, “Visualization Model of Big Data Based on Self-Organizing Feature Map Neural Network and Graphic Theory for Smart Cities,” *Cluster Computing*, vol. 22, no. 6, pp.13293-13305, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Zhibin Chen et al., “Path Controlling of Automated Vehicles for System Optimum on Transportation Networks with Heterogeneous Traffic Streams,” *Transportation Research Part C: Emerging Technologies*, vol. 110, pp.312-329, 2020.[CrossRef] [Google Scholar] [Publisher Link]

[4] Mazhar Iqbal et al., "A Fast and Reliable Dijkstra Algorithm for Online Shortest Path," *SSRG International Journal of Computer Science and Engineering*, vol. 5, no. 12, pp. 24-27, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Prashanth Venkatraman, and Michael W. Levin, “A Congestion-Aware Tabu Search Heuristic to Solve the Shared Autonomous Vehicle Routing Problem,” *Journal of Intelligent Transportation Systems*, vol. 25, no. 4, pp. 343-355, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ertugrul Bayraktar et al., “Traffic Congestion-Aware Graph-Based Vehicle Rerouting Framework from Aerial Imagery,” *Engineering Applications of Artificial Intelligence*, vol. 119, p.105769. [CrossRef] [Google Scholar] [Publisher Link]

[7] Seyyed-Mahdi Hosseini-Motlagh, Maryam Farahmand, and Mina Nouri-Harzvili, “A Multi-Path Traffic-Covering Pollution Routing Model with Simultaneous Pickup and Delivery,” *Computers & Industrial Engineering*, vol. 173, p.108644, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Dr. Shaveta Bhatia, "Survey of Shortest Path Algorithms," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 11, pp. 33-39, 2019. [CrossRef] [Publisher Link]

[9] Carise E. Schmidt et al., “Time-Dependent Fleet Size and Mix Multi-Depot Vehicle Routing Problem,” *International Journal of Production Economics*, vol. 255, p.108653, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Tianyu Zhao et al., “RNE: Computing Shortest Paths Using Road Network Embedding,” *The VLDB Journal*, vol. 31, pp. 504-528, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Dian Ouyang et al., “Efficient Shortest Path Index Maintenance on Dynamic Road Networks with Theoretical Guarantees,” *Proceedings of the VLDB Endowment*, vol. 13, no. 5, pp. 602-615, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] R.Aaditya Mehra, and G.Charan Yadav, "Implementation of Novel Routing Algorithm for Updating the Road Network Data Instantaneously," *SSRG International Journal of Geoinformatics and Geological Science*, vol. 1, no. 1, pp. 10-13, 2014. [CrossRef] [Publisher Link]

[13] Lingxiao Li et al., “Continuously Monitoring Alternative Shortest Paths on Road Networks,” *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2243-2255, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mingyu Pi et al., “Visual Cause Analytics for Traffic Congestion,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2186-2201, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sheema, D et al., “An Algorithm for Detection and Identification of Infestation Density of Pest-Fall Armyworm in Maize Plants Using Deep Learning Based on IoT,” *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 240-251, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Vaibhavi Patel, and Prof. Chitrabaggar, "A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm for Finding Shortest Path in GIS Application,” *International Journal of P2P Network Trends and Technology*, vol. 4, no. 1, pp. 21-23, 2014. [Publisher Link]

[17] Jianzhong Qi et al., “A Graph and Attentive Multi-Path Convolutional Network for Traffic Prediction,” *IEEE Transactions on Knowledge and Data Engineering*, p.1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mengxuan Zhang et al., “Stream Processing of Shortest Path Query in Dynamic Road Networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2458-2471, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zetian Jiang, Tianzhe Wang, and Junchi Yan, “Unifying Offline and Online Multi-Graph Matching via Finding Shortest Paths on Supergraph,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3648-3663, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Dhanushree.V et al., "Traffic Control and Vehicle Tracking Using RFID," *SSRG International Journal of Electronics and Communication Engineering*, vol. 5, no. 12, pp. 6-9, 2018. [CrossRef] [Publisher Link]

[21] Ranwa Al Mallah, Alejandro Quintero, and Bilal Farooq, “Prediction of Traffic Flow via Connected Vehicles,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 264-277, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Jiankun Wang et al., “Robot Path Planning via Neural-Network-Driven Prediction,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 3, pp. 451-460, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] He-xuan Hu et al., “Multi-Source Information Fusion Based Dlaas for Traffic Flow Prediction,” *IEEE Transactions on Computers*, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Sukhuman Rianthong et al., "The Reduction of Transportation Cost in Transportation Company by Evolutionary Methods," *International Journal of Engineering Trends and Technology*, vol. 69, no. 2, pp. 118-125, 2021. [CrossRef] [Publisher Link]

[25] Xiaohuan Liu et al., "Novel Best Path Selection Approach Based on Hybrid Improved A* Algorithm and Reinforcement Learning," *Applied Intelligence*, vol. 51, pp. 9015-9029. [CrossRef] [Google Scholar] [Publisher Link]

[26] Lamia Karim, Abdellah Daissaoui, and Azedine Boulmakoul, "Robust Routing Based on Urban Traffic Congestion Patterns," *Procedia Computer Science*, vol. 109, pp. 698-703, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Yuan Gao et al., "Traffic Speed Forecast in an Adjacent Region Between a Highway and Urban Expressway: Based on MFD and GRU Model," *Journal of Advanced Transportation*, vol. 2020, pp.1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] C.Ambika, M.Karnan, and R.Sivakumar, "Resolving Dynamic Shortest Path Routing Problems in Mobile Adhoc Networks Using ABC and ACO," *International Journal of Computer & Organization Trends*, vol. 3, no. 1, pp. 58-62, 2013. [Publisher Link]

[29] Yan Zheng et al., "Real-Time Prediction and Navigation on Traffic Congestion Model With Equilibrium Markov Chain," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[30] Yunchuan Sun et al., "Discovering Time-Dependent Shortest Path on Traffic Graph for Drivers towards Green Driving," *Journal of Network and Computer Applications*, vol. 83, pp. 204-212, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[31] Junhua Zhang et al., "Efficient Label-Constrained Shortest Path Queries on Road Networks: A Tree Decomposition Approach," *Proceedings of the VLDB Endowment*, vol. 15, no. 3, pp. 686-698, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] R.U.Yawle et al., "Smart Traffic Control System," *SSRG International Journal of Electronics and Communication Engineering*, vol. 3, no. 3, pp. 20-23, 2016. [CrossRef] [Publisher Link]

[33] Nils Werner, and Tim Zeitz "Combining Predicted and Live Traffic with Time-Dependent A* Potentials," *Leibniz International Proceedings in Informatics*, vol. 244, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[34] Wei Yin, and Xiaoguang Yang, "An Astar-Based Multi-Path Algorithm for the Recognition of Reasonable Route Sets in-Vehicle Navigation Systems," *Procedia-Social and Behavioural Sciences*, vol. 96, pp.1069-1078, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[35] Ms.S.Supraja, and Dr.P.Ranjith Kumar, "An Intelligent Traffic Signal Detection System Using Deep Learning," *SSRG International Journal of VLSI & Signal Processing*, vol. 8, no. 1, pp. 5-9, 2021. [CrossRef] [Publisher Link]

[36] Changshi Liu et al., "Time-Dependent Vehicle Routing Problem with Time Windows of City Logistics with a Congestion Avoidance Approach," *Knowledge-Based Systems*, vol. 188, p.104813, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[37] Mun Chon Ho et al., "An Improved Pheromone-Based Vehicle Rerouting System to Reduce Traffic Congestion," *Applied Soft Computing*, vol. 84, p.105702, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[38] Sayed Ahmed et al., "Mobile-Based Routes Network Analysis for an Emergency Response Using an Enhanced Dijkstra's Algorithm and AHP," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 6, pp. 252-260. [CrossRef] [Google Scholar]

[39] LS Jabbar et al., "A Modification of Shortest Path Algorithm According to Adjustable Weights Based on Dijkstra Algorithm," *Engineering and Technology Journal*, vol. 41, no. 2, pp. 359-374, 2023. [CrossRef] [Google Scholar] [Publisher Link]