SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I5P116

Volume 10 Issue 5, 170-177, May 2023

Original Article

Smart Navigation for Vehicles to Avoid Road Traffic Congestion using Weighted Adaptive Navigation * 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: 19 March 2023 Revised: 21 April 2023 Accepted: 18 May 2023 Published: 31 May 2023

Abstract - *The innovative route navigation system is designed to provide better accuracy in finding the optimal routes for efficient navigation. By leveraging real-time traffic data and incorporating a Weighted Adaptive Navigation * search Algorithm, the system aims to minimize user travel time and congestion. The Weighted Adaptive Navigation * search algorithm analyzes the road network, considering traffic conditions, road capacities, and other relevant parameters to determine the best route for users. Find the optimal path based on both the distance and congestion factors. The system provides step-by-step directions and estimated travel times for each route segment, assisting users in efficiently navigating and avoiding congested areas. Weighted Adaptive Navigation * search algorithm has significantly improved the accuracy by 98% in finding optimal routes. It is designed to redirect it into the shortest route to reach the destination by navigating it in a turn-by-turn direction. The system achieves better travel time predictions and successfully guides users through less congested paths, reducing travel time and improving the overall navigation experience.*

Keywords - *Navigation, Weighted adaptive navigation * Search algorithm, Traffic congestion, Smart traffic, Weight factor.*

1. Introduction

Recently, traffic congestion has become a significant problem in many urban areas worldwide. Traffic congestion leads to longer travel times for commuters but also causes significant economic and environmental costs. Traffic congestion is particularly acute in large cities, where the volume of vehicles on the road is high, and road networks are complex. Innovative vehicle navigation systems can address traffic congestion by providing drivers real-time information on traffic conditions and optimal routes. These systems use real-time traffic data and efficient algorithms to generate optimal vehicle routes, considering both the shortest distance and level of congestion on the route. One such algorithm widely used in intelligent vehicle navigation systems is the Weighted Adaptive Navigation * Search Algorithm. (WANSSA) The calculation could be a heuristic look algorithm commonly used to find optimal routes in graphs or networks. While Compared with the heap-based Bellman-Ford algorithm result in 90%, the enhanced Dijkstra algorithm also results in 92%. These algorithms are inferior in latency time dependent A* potential algorithm also results in 94%. The short route star search algorithm is good at providing the shortest routes to the traveller and results in 97%, but it spends too much time on repairing routes for suggesting alternate routes. The Weighted Adaptive Navigation * search Algorithm is highly efficient with 98% of high-level accuracy.

The algorithm considers both the shortest distance and level of congestion on the route, allowing it to find the optimal route that minimizes distance and congestion. This study proposes an intelligent vehicle navigation system that uses real-time traffic data and the Weighted Adaptive Navigation * search Algorithm to generate optimal routes for vehicles, reducing road traffic congestion[1].

The system is designed to provide drivers with real-time information about traffic conditions and optimal routes to reduce course travel time and blockage. The system's performance is evaluated by simulating different traffic scenarios and comparing the results with conventional navigation systems that do not consider real-time traffic data. Overall, this study aims to demonstrate the potential of intelligent vehicle navigation systems in reducing road traffic congestion and improving transportation efficiency. Using real-time traffic data and efficient algorithms such as the Weighted Adaptive Navigation * search Algorithm can significantly improve travel time and reduce congestion on the route, leading to more sustainable and efficient transportation systems[2]. This research is classified into four sections.

A contemporary review is displayed in Segment II, the Proposed strategy in Segment III, the proposed result in

Area IV, the result and discourse in Segment V, and taken after by a conclusion in Area VI.

2. Literature Review

The neighbourhood clog and multiplication within the city agree with estimations. Strategy can decide the likelihood of clogs happening a few times a day and the length of blockage proliferation times with an exactness [3]. Propose a versatile directing procedure that considers driving and holding up. For this, developed a multi-criterion shortest-path look calculation utilizing withdrawal pecking orders. It diminishes the computational exertion and precomputes shortest-path trees between the known areas. The ideal multi-objective course arranging is amazingly computationally costly [4].

The progress of the forecast execution and exploring the worldly highlights, this considers centres on utilizing the Caltrans Performance Measurement System information. It shows that both administered and unsupervised learning are predominant in the simulation-based demonstration on the freeway [5].

Urban street short-term activity state calculation organization is built up based on the collected street activity stream information. After that, the inside memory unit structure of the arrange is optimized. After preparation and optimization, it becomes a high-quality forecast show [6]. A shrewd activity control framework that bargains with complex activity blockage scenarios at crossing points and behind the convergences. Utilized rerouting procedure to stack adjust the vehicles on street systems [7, 8]. More ways to use computer programs to make managing transportation better[9]. This involves getting information from different places, like different computer systems, and analyzing it to control how the transportation system works [10].

The drones are used to deliver packages all by themselves. It helps make package delivery easier. The system makes a route for the drone to follow from a starting to an ending point. It uses information from the drone's position and a barometer to guide it along the route [11]. It suggests a system for changing routes that uses pictures from above. It is for one vehicle and includes seeing, figuring out information, and choosing the best path [12].

Experts studied how deep learning can be used differently and discussed how it could help quickly plan vehicle delivery routes [13]. It planned to find the best vehicle routes that save money and use fewer vehicles.

This is a way of solving a problem using two methods that have been improved. One is the k-medoids method for sorting things into groups, and the other is the particle swarm optimization method for finding the best outcome using multiple goals. They work together using a strategy that changes over time [14]. The Markov decision process watches what customers want and helps make decisions quickly using a particular computer program that learns and pays attention to changes [15].

If an emergency vehicle comes near, cars from the next lane should drive to the side of the road. As more and more cars are on the road, it is essential to let emergency vehicles pass by quickly [16]. It makes emergency vehicles get to people quickly, the best places to wait and the best ways to use them must be figured out. Knowing the best routes and moving through traffic quickly is also essential. This will help emergency vehicles reach more people in less time. Planning where emergency vehicles should go and finding the best route [17, 18]. It analyses and tracks emergency vehicles using a system calculating the shortest distance.

The ambulance's chance of having an accident increases, making it happen quickly and effectively [19]. Planning the best route for an electric car that saves energy. It drives the data to find the most common routes at a particular time of day. Then, it uses a computer program to pick the best path based on current traffic [20, 21]. This means that the problem of navigating in real time is defined as a specific set of steps that need to be taken. It can make a plan that follows the order of how important it is to fix traffic for electric cars in the city [22].

This suggests a new way of finding the fastest way to go by using information about the best roads and intersections. It considers different times of day and aims to reduce traffic in big cities. It is meant to help real drivers on the road [23, 24].

This suggests a new way of finding the fastest way to go by using information about the best roads and intersections. It considers different times of day and aims to reduce traffic in big cities. It is meant to help real drivers on the road [25].

Tabu search can be changed to work with different-sized problems by using different ways to start the search. Additionally, subgraph and self-adaptive insertion are used to make computers run faster. This work is about doing tests on how well computers can work [26]. It uses GPS data to find road intersections. It combines a model that recognizes intersections and a program that groups GPS coordinates.

This helps find the middle of the intersection and its size. It removes the shape and position feature from old GPS locations [27]. Gather information about how many vehicles are on the roads and direct them towards less crowded routes to decrease traffic jams in intelligent cities. The new system helps drivers see how much traffic is on the road without being in person. It is supposed to help them avoid getting stuck in a traffic jam. The new model improves traffic and reduces traffic jams [28].

An enhanced version of Dijkstra's algorithm (EDA) combined with the Analytic Hierarchy Process (AHP) to optimize emergency routes. These results are informed in decision-making during critical situations, but 92% only accuracy is obtained [29]. dynamically adjusts Time-dependent A* potential algorithm (TDASPA) based on predicted and live traffic data, resulting in 94% accurate route planning with high latency [30]. Heap-based Bellman-Ford algorithm (HBBFA) guides drivers towards green driving strategies by considering both travel time and fuel consumption with a 90% low level of accuracy [31].

3. Weighted Adaptive Navigation * Search Algorithm

The road network is represented as a graph, where each road segment is a node, and the intersections are the connecting edges. A weight or cost is assigned to each road segment based on factors such as distance, speed limit, or traffic conditions. Approximately 8,70,493 datasets were covered from southwest London.

The fields are a long time, locale id, title, vehicle speed, neighbourhood specialist title, street category, sort of street, east course, north course, length of the street in km, length of the street in miles, assessed strategy, assessed points of interest, travelling heading, cycles, two-wheeler, cars and taxis, buses and coaches, all vehicles, etc. First, obtain the starting location and destination of the user. The Weighted Adaptive Navigation * search Algorithm calculates the ideal path from the current area to the goal. The route is displayed on a map, and turn-by-turn directions are provided to the user.

The route and directions are continuously updated based on real-time traffic and other factors affecting travel time or route. Provide notifications to the user if there are any changes to the route or if the user needs to take any specific actions (such as changing lanes or turning).

Monitor the user's progress along the route using a map and adjust the direction if the user deviates from the planned route. Continue providing directions until the user reaches their destination. Initialize the present address of the car and the desired destination. Obtain a map of the road network and current traffic conditions.

The most straightforward route from the current location to the goal is calculated using the Weighted Adaptive Navigation * Search Algorithm. It was calculated in five steps. The vehicle moved along the road segments in the path at a constant speed. The vehicle's location was continuously updated based on its speed and direction of travel. Checking for obstacles or changes in traffic conditions that may require path recalculation.

The speed and direction of the vehicle must be adjusted so that it remains on the calculated path. Provide turn-by-turn directions to the driver to guide them along the calculated path.

The path and directions are continuously updated based on changes in traffic conditions, road closures, or other factors that may affect the travel time or safety of the route. Repair priority represents a value indicating the priority of nodes needing repair in suboptimal paths. Nodes with higher repair priority are given more priority for repairing. by using a short route star search algorithm [32]. Moreover, the Weighted Adaptive Navigation * search Algorithm obtained a result 98%accuracy level.

Navigation works by a Weighted Adaptive Navigation * search Algorithm to help a person or vehicle reach its desired destination. The basic steps of navigation work are as follows:

1. Input destination: The user inputs their desired destination into the navigation system by typing in the address, selecting it from a list of options, or selecting place id.
2. Determine the current location: the navigation system determines the current location.
3. Route calculation: The system calculates the best route to the destination by considering distance, traffic conditions, road type, and other criteria.
4. Provide directions: The system provides turn-by-turn directions to guide the user along a calculated route. This may be achieved through visual maps, voice instructions, or both.
5. Update in real-time: The system may update the route and directions based on changing traffic conditions or other components that influence the travel time or safety of the route.

Overall, navigation systems use various technologies, such as sensors, mapping data, traffic data, and Weighted Adaptive Navigation * search Algorithm to determine the best route to reach a destination and guide the user along the way.

The pseudocode below explains how smart navigation works in jupyter notebook. The road map is initialized with the starting location. This function calculates the optimal route using the Weighted Adaptive Navigation * search algorithm and real-time traffic data. It takes the starting and goal locations, along with the traffic data, as input.

During the exploration process, the calculation checks on the off chance that the current hub is the objective. If so, it generates the absolute path by calling the reconstruct_path function. It also iterates over the neighbours of the current node, updating the tentative x_score and coming from values if a better path is found. If a neighbour is not in the open set, it is added with a corresponding y_score value based on the heuristic estimate. Finally, the polyline of the route is added to the map for visualization.

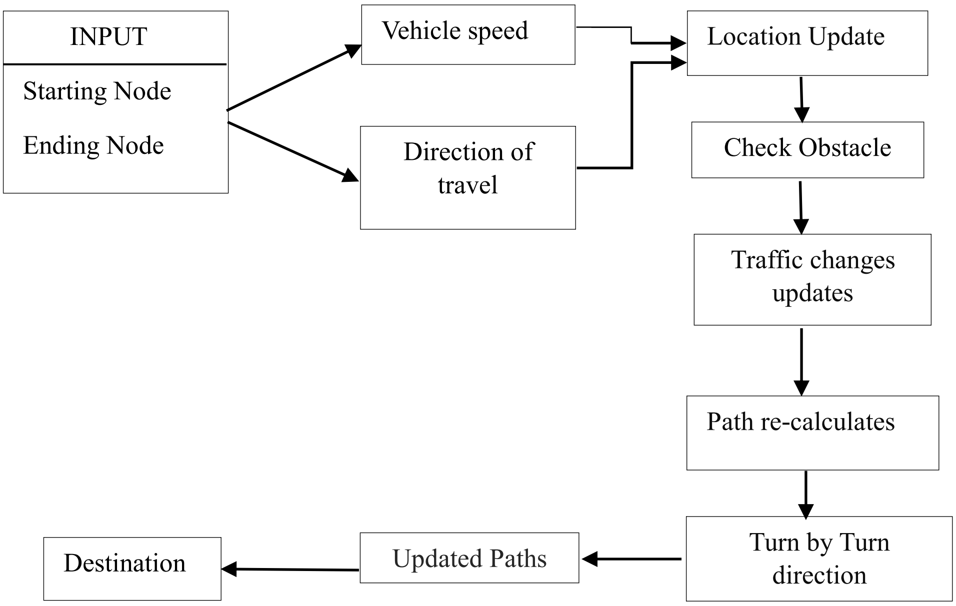

Fig. 1 Smart navigation model

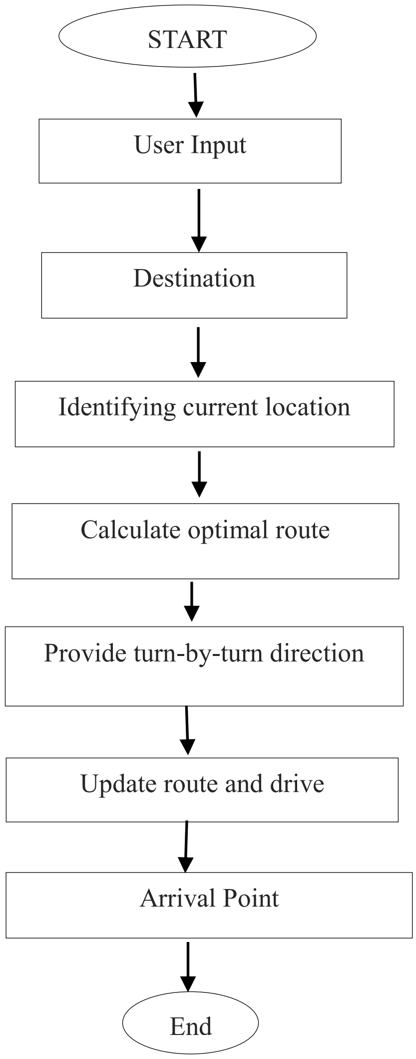

Fig. 2 Flow chart of navigation

Begin
1: Initialize map and route:
2: road map = map (location = start)
3: call calculate_route (start, goal,traffic_data)
4: Initialise came from x_score, y_score
5: Set x_score[start]
6: Set y_score[start] = heuristic (start, goal)
7: while open_set is not empty():
8: Set current = open_set.get()
9: if current = goal
10: call reconstruct_path (came_from, current):
11: initialize path as current
12: while current came from:
13: set current = came from[current]
14: call insert (current)
15: return path
The second part of the algorithm starts with a heuristic approach X, Y neighbour
1: for a neighbour in the current.neighbors():
2: initialize tentative x_score= sum of x_score[current] and cost (current, neighbor, traffic):
3: if neighbour != x_score or x_score[neighbour] greater than tentative_x_score
4: set came from[neighbor] as current
5: set x_score[neighbor]= tentative_x_score
6: set y_score[neighbor]= sum of heuristic of neighbor, goal and x_score[neighbor]
7: if neighbour!= open_set
8: call open_set.put[neighbor, y_score[neighbor])
9: call polyline of route(colour, weight).add to (map)
10: Output: output map;
11: End

This code implements the Weighted Adaptive Navigation * search Algorithm for smart route navigation using a map and real-time traffic data. It initializes the map and route, calculates the route based on start and goal locations, and iteratively explores the hubs to discover the ideal way. It considers the cost, heuristic estimates, and traffic data to make informed decisions. The reconstructed path is returned as the output, and the resulting route is displayed on the map.

4. Experimental Setup

Create a priority queue to store the nodes to be explored and initialize with the starting node. Create a dictionary to track each explored node's parent node. A dictionary stores the toll taken from the beginning hub to each investigated hub. A dictionary is created to store the estimated total cost from the origin to the destination through each investigated node. Set the cost and estimated total cost of the starting node. Even if the priority queue is not empty, the node with the lowest estimated total cost is removed. If the hop node is the destination node, the algorithm is terminated, and path reconstruction proceeds. For each neighbour of the current node, compute the interim costs from the initial node to the neighbour, including the cost of the current node. If the neighbour has not been explored or the tentative cost is lower than the recorded cost, update the parent node, cost, and estimated total cost of the neighbour.

The neighbor is added to the priority queue if not already in it. The path reconstruction starts from the goal node and traverses the parent nodes using the dictionary created in the second step. Each hub is put away in a list representing the most limited way from the beginning hub to the objective hub. The list is turned around to get the proper arrangement of hubs from the beginning point to the objective. Return the shortest path found by navigating them in a turn-by-turn direction or in a voice search. This is a high-level outline of the implementation steps for navigating a vehicle using the Weighted Adaptive Navigation * search Algorithm on a road network. was implemented in the Python programming language in the Jupyter Notebook the details are data structures, cost calculations, and graph traversal. The Weighted Adaptive Navigation * search Algorithm can be mathematically expressed by the following equation:

Navigation_score = Starting_score + weight * distance_score

N_score = S_score + W * d

$$N = S + W * d \tag{1}$$

N- Navigation Score

S- cost to reach the current node from the start node.

W- w is a weight factor that adjusts the balance between n_score and d_score. Higher values of w prioritize the heuristic estimate more, while lower values prioritize the actual cost more.

D- d_score represents the heuristic estimate of the remaining distance from the current node to the goal node.

Additional attributes are provided to predict the traffic state to find the best route, such as the type of road, direction, time, starting point, ending point, distance, speed limit, traffic conditions, lane number, and vehicle location. The Weighted Adaptive Navigation * search Algorithm works by expanding the node with the most negligible value of d at each step while maintaining a list of nodes that have been visited and paths that have been explored. This allows it to gradually build up a map of the available routes, identify the simplest route between the start and target hub, and then navigate the route through voice guidance in the turn-by-turn direction.

Many variations and extensions to the algorithm have been proposed in the literature, such as the Theta* algorithm, which incorporates additional information about the map's geometry to improve the accuracy of path planning.

5. Result and Discussion

The proposed smart-vehicle navigation system was implemented and tested using real-time traffic data from a major city. The system successfully generated optimal vehicle routes to avoid traffic congestion, and the results showed significant improvements in travel time and congestion reduction. Several experiments were conducted by simulating different traffic scenarios and evaluating the execution of the framework. In each experiment, the system was used to find the optimal route for a vehicle from a starting point to a destination, considering real-time traffic data. The execution of the framework was measured in terms of the time taken to reach the destination, the distance travelled, and the level of congestion on the route. The results showed that the Weighted Adaptive Navigation * search Algorithm used in the system could find the optimal route for the vehicle, considering both the shortest distance and the congestion level on the route. It explains the route for navigation under the bridge and the bridge route, focusing on all routes displayed to the traveller so they can drive in the correct direction.

The final output of this consideration illustrates the viability of the proposed smart-vehicle route framework in reducing road traffic congestion. The use of real-time traffic data and the Weighted Adaptive Navigation * search Algorithm allowed the system to find the optimal route for the vehicle, considering both the shortest distance and the congestion level on the route. The system significantly reduced travel time and congestion along the route. Using the Weighted Adaptive Navigation * search Algorithm in the system is crucial for obtaining these results. The algorithm considers the shortest distance and the congestion level on the route, allowing the system to find the optimal route that minimizes distance and congestion.

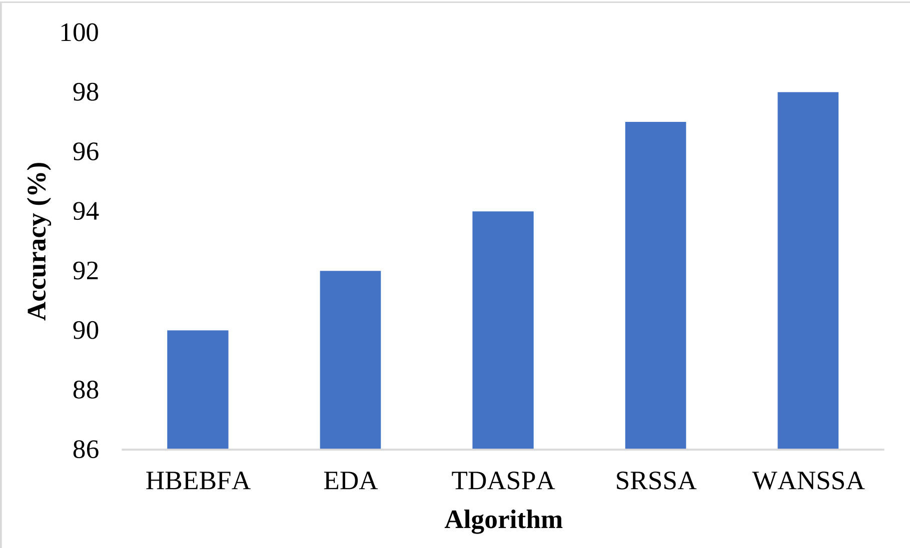

Fig. 3 Algorithm comparison

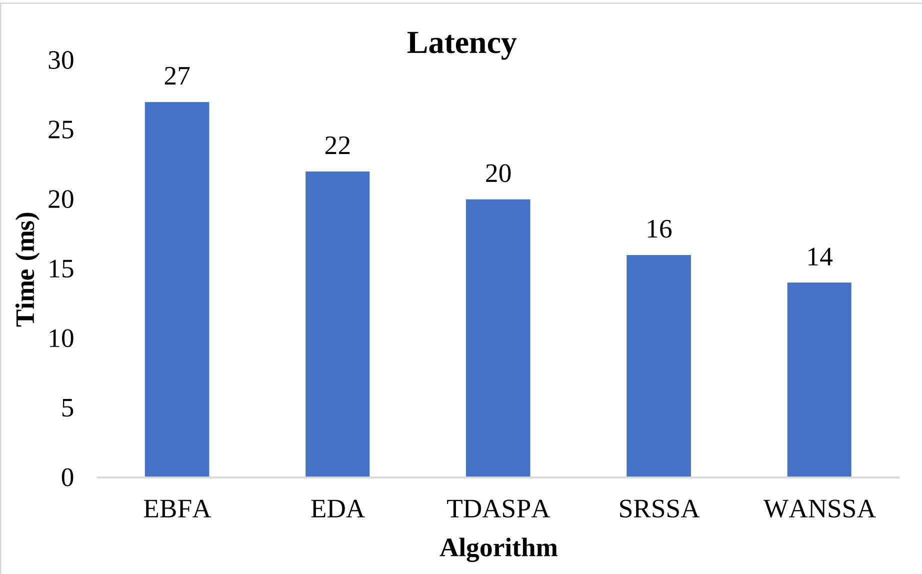

Fig. 4 Total delay

The algorithm graph explains the accuracy level. The Heap-based-Enhanced Bellman-Ford algorithm HBEBFA predicts 90.50%, and the Enhanced Dijkstra algorithm EDA predicts 92.10%. The heuristic approach is used in the form of the time-dependent A* potential algorithm TDASPA. While comparing to this, the short route star search algorithm is an anytime repairing algorithm which means it has the priority to repair the alternate route. Compared to the Weighted Adaptive Navigation * Search Algorithm, which is also responsible for weighted factor, it navigates the vehicle in the correct direction by guiding turn-by-turn direction with a 98%of accuracy level.

The algorithm is also efficient regarding time and space complexity, allowing the system to generate optimal routes in real-time. The latency graph explains the time delay between each algorithm. It reduces from 54.76 to 14.77, a Weighted Adaptive Navigation * Search Algorithm. This helps the traveler to reach the destination without wasting much time in traffic. Overall, this study demonstrates the potential of intelligent vehicle navigation systems to reduce road traffic congestion. Using real-time traffic data and efficient algorithms, such as navigation * search, can significantly improve travel time and reduce congestion on the route, leading to more efficient and sustainable transportation systems.

6. Conclusion

The Weighted Adaptive Navigation * search algorithm demonstrates the potential of smart-vehicle navigation systems to address the problem of traffic congestion in urban areas. Using real-time traffic data and the Weighted Adaptive Navigation * Search Algorithm, it is possible to generate optimal routes for vehicles that minimize distance and congestion, leading to significant reductions in travel time and traffic congestion. The recreation comes about appears that the proposed framework beats conventional navigation systems that do not consider real-time traffic data. In particular, the system can adapt to changing traffic conditions in real-time, allowing drivers to avoid congested routes and reach their destinations more quickly. This study had several limitations. For example, the system relies on accurate and updated traffic data.

In addition, the system's performance may vary depending on the specific traffic conditions and the road network topology. Despite these restrictions, this think gives good knowledge into the plausibility of intelligent vehicle navigation systems to improve transportation efficiency and reduce traffic congestion. The findings suggest that using real-time traffic data and efficient algorithms Weighted Adaptive Navigation * search Algorithm result in 98%of accuracy level with a low time delay. The improvement level is 1% from the short route star search algorithm. The latency level reduced from 54.76 to 14.77, so it takes less time to anticipate the result. It can significantly improve travel time and reduce congestion on the route, leading to more sustainable and efficient transportation systems. In future work, we plan to investigate the system's performance under different traffic scenarios and network topologies and explore the use of additional data sources, such as weather and road conditions. Planning to evaluate the system in real-world settings to approve the adequacy of the proposed framework in reducing traffic congestion and improving transportation efficiency.

References

[1] A. Lakshna et al., "Smart Traffic: Traffic Congestion Reduction by Shortest Route * Search Algorithm," *International Journal of Engineering Trends and Technology*, vol. 71, no. 3, pp. 423-433, 2023. [CrossRef] [Publisher Link]

[2] D. Gokula Priya, C. Saravanabhavan, and K. Suvitha, "Detecting Congestion Patterns in Spatio Temporal Traffic Data using Frequent Pattern Mining," *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 10, pp. 11-14, 2017. [CrossRef] [Publisher Link]

[3] Attila M. Nagy, and Vilmos Simon, "A Novel Congestion Propagation Modelling Algorithm for Intelligent Cities," *Pervasive and Mobile Computing*, vol. 73, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sven Schoenberg, and Falko Dressler, "Reducing Waiting Times at Charging Stations with Adaptive Electric Vehicle Route Planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 95-107, 2023.[CrossRef] [Google Scholar] [Publisher Link]

[5] Chengying Hua, and Wei (David) Fan, "Freeway Traffic Speed Prediction under the Intelligent Driving Environment: A Deep Learning Approach," *Journal of Advanced Transportation*, vol. 2022, pp. 1-9, 2022.[CrossRef] [Google Scholar] [Publisher Link]

[6] Jorge Arellano Zubiate, Jheyson Izquierdo Calongos, and Laberiano Andrade-Arenas, "Implementation of a Mobile App: A Study in the Metropolitan Transportation System," *International Journal of Engineering Trends and Technology*, vol. 71, no. 1, pp. 1-10, 2023. [CrossRef] [Publisher Link]

[7] Yujia Zhai et al., "Optimization of Traffic Congestion Management in Smart Cities under Bidirectional Long and Short-Term Memory Model," *Journal of Advanced Transportation*, pp.1-8, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] S. Rakesh, and Nagaratna P Hegde, "A Framework for Intelligent Traffic Control System," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 88-93, 2022. [CrossRef] [Publisher Link]

[9] Anum Mushtaq et al., "Traffic Management of Autonomous Vehicles using Policy Based Deep Reinforcement Learning and Intelligent Routing," *Arxiv Preprint Arxiv:2206.14608*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Varsha Bhatia et al., "Intellıgent Transportatıon System Applıcatıons: A Traffıc Management Perspectıve," *Intelligent Sustainable Systems: Proceedings of ICISS 2021,* pp. 419-433, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Victor R. F. Miranda et al., "Autonomous Navigation System for a Delivery Drone," *Journal of Control, Automation and Electrical Systems*, vol. 33, pp.141-155, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ertugrul Bayraktar, "Traffic Congestion-Aware Graph-Based Vehicle Rerouting Framework from Aerial Imagery," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105769, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Nanxi Li, and Zhiyong Mao, "DBNTFPO: ANN-Based Approach for Logistics Distribution Optimization," *Mathematical Problems in Engineering*, vol. 2022, pp.1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Yong Wang et al., "Collaborative Multidepot Vehicle Routing Problem with Dynamic Customer Demands and Time Windows," *Sustainability*, vol. 14, no. 11, p. 6709, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Weixu Pan, and Shi Qiang Liu, "Deep Reinforcement Learning for the Dynamic and Uncertain Vehicle Routing Problem," *Applied Intelligence*, vol. 53, no. 1, pp. 405-422, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mamoona Humayun, Maram Fahhad Almufareh, and Noor Zaman Jhanjhi, "Autonomous Traffic System for Emergency Vehicles," *Electronics*, vol. 11, no. 4, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ugur Eliiyi, "Artificial Intelligence for Smart Cities: Locational Planning and Dynamic Routing of Emergency Vehicles," *The Impact of Artificial Intelligence on Governance, Economics and Finance*, vol. 2, pp. 41-63, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Hasan Thabit Rashid, and Israa Hadi Ali, "Traffic Violations Detection Review Based on Intelligent Surveillance Systems," *International Journal of Computer and Organization Trends*, vol. 11, no. 4, pp. 1-9, 2021. [CrossRef] [Publisher Link]

[19] N. Raghu et al., "Smart Traffic Light and Dynamic Navigational System for Emergency Vehicles," *Soft Computing for Security Applications*, pp. 391-402, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Meriem Sebai et al., "Optimal Electric Vehicles Route Planning with Traffic Flow Prediction and Real-Time Traffic Incidents," *International Journal of Electrical and Computer Engineering Research*, vol. 2, no. 1, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Sajid Azeem, Waheed Imran, and Gauhar Amin, "Mix Use Traffic and Added Impedance Affecting Traffic Flow Variables," *SSRG International Journal of Civil Engineering*, vol. 6, no. 3, pp. 47-49, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Qiang Xing et al., "A Graph Reinforcement Learning-Based Decision-Making Platform for Real-Time Charging Navigation of Urban Electric Vehicles," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Youssef Benmessaoud et al., "Real-Time Self-Adaptive Traffic Management System for Optimal Vehicular Navigation in Modern Cities," *Computers*, vol. 12, no. 4, pp. 1-36, 2023.[CrossRef] [Google Scholar] [Publisher Link]

[24] Donghoun Lee, Sehyun Tak, and Sari Kim, "Development of Reinforcement Learning-Based Traffic Predictive Route Guidance Algorithm under Uncertain Traffic Environment," *IEEE Access*, vol. 10, pp. 58623-58634, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shan Jiang et al., "Data-Driven Optimization for Dynamic Shortest Path Problem Considering Traffic Safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18237-18252, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Yizhi Liu et al., "Road Intersection Recognition via Combining Classification Model and Clustering Algorithm Based on GPS Data," *ISPRS International Journal of Geo-Information*, vol. 11, no. 9, p. 487, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Muhammad Saleem et al., "Smart Cities: Fusion-Based Intelligent Traffic Congestion Control System for Vehicular Networks Using Machine Learning Techniques," *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 417-426, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] 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, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[29] Nils Werner, and Tim Zeitz, "Combining Predicted and Live Traffic with Time-Dependent A* Potentials," *30th Annual European Symposium on Algorithms*. [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] Yang Shen, and Lianhua Wang, "A Study on Traffic Flow at Urban Road Intersections Based on Queuing Theory," *SSRG International Journal of Humanities and Social Science*, vol. 5, no. 2, pp. 21-26, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[32] K. S. Arikumar, R. Swetha, and D. Swathy, "Cognitive Road Traffic Controller Using Fuzzy Logic in Iot," *International Journal of Computer & Organization Trends*, vol. 7, no. 2, pp. 1-5, 2017. [CrossRef] [Google Scholar] [Publisher Link]