UKR Journal of Multidisciplinary Studies (UKRJMS)
Homepage: https://ukrpublisher.com/ukrjms/
Email: submit.ukrpublisher@gmail.com
Volume 2, Issue 4, 2026
ISSN: 3049-432X (Online)
OPEN ACCESS

Optimization of Vehicular Traffic Systems Using Modified Shortest Path Algorithm Within Calabar Metropolis

Solomon, F. U.[1]; J. Obidinnu[2]; Ibe, W.E.[3]; S.O.O, Duke [4]*

[1,2,4] Department of Computer Science, University of Cross River State, NIGERIA.
[3]Department of Computer Science, Federal College of Education Obudu, NIGERIA.

***Corresponding Author**: S.O.O, Duke
DOI: *https://doi.org/10.5281/zenodo.19666342*

Article History
Original Research Article
Received: 01-03-2026
Accepted: 06-04-2026
Published: 20-04-2026

Citation: Solomon, F. U.; J. Obidinnu; Ibe, W.E.; S.O.O, Duke. (2026). Optimization Of Vehicular Traffic Systems Using Modified Shortest Path Algorithm Within Calabar Metropolis. UKR Journal of Multidisciplinary Studies (UKRJMS), 2(4), 89-106.

Abstract

Traffic congestion remains a pressing challenge in many urban areas, particularly within growing cities such as Calabar Metropolis. The increase in vehicle density on limited road networks often leads to delays, fuel wastage, and reduced commuter productivity. Traditional shortest path algorithms provide efficient distance-based routing but fail to adequately capture the impact of congestion on travel time and cost. This limitation highlights the need for a more dynamic and adaptive solution that integrates traffic conditions into route computation. This research aimed at developing a vehicular traffic optimization system using a modified Dijkstra shortest path algorithm. The system was designed to compute optimal paths within Calabar's major road networks, dynamically adjusting edge weights to reflect congestion scenarios. Implementation was carried out using we development technologies, with a structured storage mechanism serving as the system's database. The interface was equipped with a "Costometer" sidebar that presented route costs and travel distances, alongside clear visualization of roads such as Watt Roundabout, Manye Avenue, White House Street, Palm Street, and their adjoining routes. In its workings, the user selects the starting point, and target location, the system compute cost in minutes. While in a heavy traffic, the system recomputes cost from the current position of the user and provided another route helping the user to escape staying long in traffic. The system also provide information to the user if it is not possible to leave the traffic based on traffic condition. Experimental testing showed that the modified algorithm effectively rerouted traffic away from congested roads while maintaining efficiency in non-congested cases. The Costometer provided real-time route cost updates, while the storage system ensured efficient retrieval of traffic data. The results demonstrate that the proposed system offers a practical and reliable tool for congestion-aware navigation and traffic optimization in Calabar Metropolis.

Keywords: *Traffic Congestion, Optimization, Modified Shortest Path Algorithm, Costometer, Navigation.*

1.0 INTRODUCTION

Road transportation plays a critical role in economic and social development, yet it continues to face persistent challenges such as traffic congestion, frequent accidents, and inefficient route utilization. In Nigeria, climatic factors such as flooding and extreme weather further contribute to the vulnerability of road networks, disrupting transportation flow and increasing travel delays [1]. Rapid urbanization and population growth have intensified these challenges, particularly in metropolitan areas, where limited infrastructure struggles to meet rising vehicular demands [2]. These realities highlight the urgency of adopting advanced traffic optimization systems to enhance safety, efficiency, and resilience of road networks.

In this current research, the proposed modification introduces a software-based traffic optimization system designed to overcome the limitations of traditional shortest path algorithms, which rely on static weights. Unlike existing systems, the new solution integrates real-time traffic data into the pathfinding process, ensuring that routing decisions reflect current road conditions. The system ingests live inputs such as average road speeds, congestion levels, and incident alerts from external sourcess or simulated feeds, which are then converted into dynamic edge weights on the traffic graph. A modified version of algorithms like Dijkstra or A* powers the dynamic pathfinding engine, continuously recalculating optimal routes whenever road conditions change, but without recomputing the entire network to save time. This enables the system to deliver real-time guidance through a user-friendly software interface, whether as a mobile, desktop, or web application, providing drivers with updated routes, alternative paths, and live traffic visualizations as conditions evolve.

Technological advancements in intelligent transportation systems (ITS) and vehicular communication networks have created opportunities for real-time traffic management and data-driven decision-making. Emerging methods leverage algorithms, artificial intelligence, and IoT-enabled devices to predict traffic conditions, guide vehicles through optimal routes, and reduce congestion [3],[4]. For instance, queueing theory-based systems and machine learning models have demonstrated potential in modeling vehicular flow and improving traffic control mechanisms [5]. Such approaches provide a foundation for developing smart traffic solutions tailored to Nigeria's unique road challenges.

In addition, optimization algorithms such as Dijkstra, A*, and Bellman-Ford have been applied extensively in solving pathfinding problems within logistics, robotics, and transportation. Their adoption in vehicular routing allows for dynamic identification of shortest or least congested paths, reducing travel time and resource consumption [6], [7]. However, while these algorithms have proven effective in controlled environments, their direct application in Nigeria faces contextual challenges such as irregular road networks, poor compliance with traffic rules, and unpredictable driver behaviors [8]. This necessitates modifications and integration with adaptive, data-driven systems capable of responding to local complexities.

Furthermore, research in traffic management increasingly emphasizes the integration of optimization with sustainability goals. Studies show that well-designed signal control systems and algorithm-based route guidance not only reduce travel delays but also minimize fuel consumption and vehicular emissions, contributing to greener urban environments [9], [10]. In Nigeria, where traffic congestion contributes significantly to economic losses and public health risks, adopting such intelligent optimization systems presents a dual opportunity: improving transportation efficiency while addressing environmental and social concerns [11]. This background sets the foundation for exploring advanced optimization models that can effectively enhance vehicular traffic systems in the Nigerian context.

The proposed system is a web-based traffic optimization platform that integrates real-time traffic data into a dynamic shortest path algorithm to improve routing efficiency. Unlike conventional systems that rely on static road weights, this solution continuously ingests live inputs—such as average road speeds, congestion indices, and incident alerts—from traffic sourcess and road databases, and translates them into updated edge weights on a traffic graph. A modified Dijkstra/A* algorithm forms the core of the pathfinding engine, recalculating only the affected nodes and edges whenever significant changes occur, thereby reducing computational overhead while ensuring accuracy. The decision layer then selects and outputs the most efficient route in real time, automatically suggesting alternative paths if traffic conditions change mid-journey. Through its web interface, the system presents drivers with an interactive map that highlights the current optimal route, traffic conditions, and possible detours, offering a user-friendly solution that adapts instantly to dynamic road environments

2.0 LITERATURE FOUNDATION

2.1 Vehicular Traffic System in Calabar Metropolise

Vehicular traffic systems form the foundation of modern transportation networks, facilitating the movement of people, goods, and services across urban and rural areas. These systems encompass a wide range of components including road infrastructures, vehicles, drivers, and communication technologies that collectively determine traffic flow efficiency. The effectiveness of a vehicular traffic system is strongly influenced by factors such as road design, traffic regulations, climatic conditions, and driver behavior [12]. In many developing regions, traffic systems often face unique challenges due to inadequate infrastructure, rapid urbanization, and limited integration of intelligent traffic management solutions.

Technological advancements have significantly transformed vehicular traffic systems over the last decade. The integration of vehicular communication networks and connected autonomous vehicles (CAVs) has enhanced real-time data sharing, enabling advanced traffic monitoring and congestion management strategies [13]. Similarly, industrial IoT and artificial intelligence (AI) have been applied in vehicular logistics and transportation systems to optimize routing, minimize delays, and improve road safety outcomes [14]. These developments indicate a shift from conventional, reactive traffic systems to proactive and predictive traffic management frameworks that leverage real-time analytics.

Despite these innovations, traffic-related issues such as congestion, accidents, and delays remain pressing concerns. Studies highlight that traffic accidents are not only influenced by infrastructural deficiencies but also by human factors such as poor driver-to-driver communication and non-compliance with road safety rules [15], [16]. Moreover, climatic factors such as extreme rainfall and heat events have been shown to exacerbate traffic vulnerabilities, particularly in regions like Nigeria, where climate change impacts road infrastructure reliability [17]. These challenges underscore the importance of developing adaptive vehicular traffic systems that account for both environmental and human behavioral dynamics.

To address these complexities, researchers have explored various optimization models. Queueing theory-based approaches, for instance, have been applied to analyze traffic flows and reduce congestion by modeling vehicles as interconnected systems within a network [18]. Similarly, traffic signal optimization and ensemble learning-based prediction models have been employed to improve traffic sustainability and predict flow dynamics in smart transportation systems [19], [20]. Collectively, these studies establish the background of vehicular traffic systems as a multidisciplinary field that combines engineering, computer science, and behavioral analysis to create efficient, safe, and resilient transportation networks.

Vehicular traffic systems in Calabar, the capital of Cross River State, reflect the broader challenges of urban traffic management in medium-sized Nigerian cities. The city's road network, though moderately developed, often experiences congestion at peak hours due to a mix of private vehicles, public buses, motorcycles, and tricycles competing for limited road space. Informal driving practices and weak enforcement of traffic regulations further compound traffic inefficiencies, leading to frequent delays and heightened risks of accidents. Seasonal flooding, a recurring problem in Calabar, also disrupts traffic flow and highlights the vulnerability of the city's road system to climate-related factors, aligning with findings that emphasize the role of environmental conditions in traffic performance [21].

On a national scale, vehicular traffic systems in Nigeria face more significant structural and operational challenges. Rapid urbanization and population growth in cities such as Lagos, Abuja, Port Harcourt, and Kano have placed enormous pressure on existing road infrastructure, resulting in chronic congestion, delays, and elevated accident rates. Inadequate integration of intelligent transportation systems (ITS), insufficient traffic monitoring facilities, and irregular road maintenance further undermine the efficiency of Nigeria's traffic systems [22], [23]. Moreover, driver-related factors including poor communication skills, disregard for traffic signals, and high levels of road rage have been identified as critical contributors to traffic accidents across the country [24].

Efforts to optimize vehicular traffic systems in Nigeria have begun to leverage both technological and policy-driven approaches. Some cities are adopting intelligent traffic signal systems and data-driven congestion monitoring tools to reduce waiting times at intersections [25]. There is also growing interest in using machine learning and ensemble-based predictive algorithms to forecast traffic patterns and guide route planning [26]. However, the widespread adoption of these technologies remains constrained by limited infrastructure, inadequate funding, and weak institutional support. Consequently, Nigeria's vehicular traffic system requires holistic reforms that combine infrastructural development, climate adaptation strategies, behavioral interventions, and technological innovation to ensure safe, efficient, and sustainable mobility.

2.2 Traffic Signal Control Systems

Traffic signal control systems are fundamental components of vehicular traffic optimization, as they regulate the flow of vehicles at intersections to minimize congestion and improve safety. In modern systems, the process follows a structured flow beginning with data input, proceeding through software-based processing and decision-making, and culminating in traffic signal output. This is shown in Figure 2.0 below;

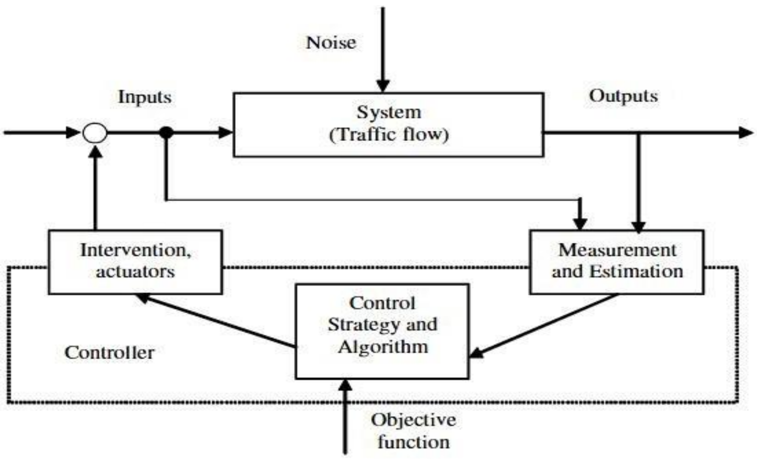

Figure 1: *Traditional Vehicular Traffic Control System [27]*

1. **Input Layer (Data Collection):**

Traffic signal systems rely on various data sources to capture real-time traffic conditions. These include inductive loop detectors embedded in roads, infrared or video cameras, radar sensors, and increasingly, connected vehicle data from vehicular communication networks [28]. The collected data typically represents traffic volume, vehicle speed, queue length, and waiting time. In advanced systems, additional contextual factors such as pedestrian movements, weather conditions, and even accident alerts may also serve as input.

2. **Processing Layer (Software and Algorithms):**

The collected input data is transmitted to the traffic control software, where it is processed using algorithms designed to optimize traffic flow. Traditional systems rely on fixed-time control based on pre-set schedules, but more advanced systems employ adaptive algorithms that adjust signal timing dynamically. For example, queueing theory and optimization models can calculate the most efficient allocation of green and red times [29]. Recent innovations use machine learning models and ensemble learning-based prediction algorithms to forecast near-future traffic states and proactively adjust signal plans [30]. The software integrates these models into decision rules that determine which traffic movement should be prioritized at any given moment.

3. **Decision Layer (Control Logic):**

Once the data has been processed, the system generates control signals based on its decision logic. For instance, if a heavy queue is detected on one approach while cross traffic is light, the software may extend the green phase for the congested direction. Similarly, if pedestrian activity is detected, the system may allocate a dedicated crossing phase. In coordinated traffic management systems, multiple intersections are linked, and the control software ensures synchronization to maintain smooth flow along major corridors [31].

4. **Output Layer (Signal Execution):**

The final stage is the transmission of commands from the control software to the physical traffic signal controllers at intersections. These controllers execute the timing changes by adjusting the duration of green, yellow, and red lights. In intelligent traffic systems, this output is continuously updated in real time, ensuring that the signals adapt dynamically to changing traffic conditions rather than operating on static cycles. The result is improved traffic throughput, reduced idle time, lower emissions, and enhanced road safety [32].

2.3 Importance of Optimization in Traffic Systems

Optimization in traffic systems is essential for improving mobility, reducing delays, and ensuring efficient utilization of limited road infrastructure. As urban populations and vehicular ownership continue to rise, cities experience increasing traffic congestion that hampers economic productivity and contributes to higher transportation costs. Optimization strategies, such as adaptive signal control, dynamic route guidance, and real-time monitoring, enable road networks to handle larger traffic volumes more effectively while minimizing gridlock [33], [34]. Without such measures, traffic systems risk becoming overstretched, resulting in longer travel times, fuel wastage, and greater driver frustration.

Beyond congestion reduction, optimization is critical for enhancing road safety. Studies reveal that poor traffic management and ineffective driver-to-driver communication significantly contribute to accidents, especially in developing regions with weak enforcement of traffic rules [35], [36]. By incorporating intelligent transportation systems (ITS), predictive analytics, and real-time decision-making models, traffic optimization frameworks can mitigate accident risks by identifying hazards early and adapting traffic flows accordingly. For instance, queueing theory-based optimization techniques ensure smoother traffic distribution, reducing the likelihood of bottlenecks and collision-prone intersections [37].

Environmental sustainability also underscores the importance of optimization in traffic systems. Prolonged congestion increases greenhouse gas emissions, deteriorates air quality, and exacerbates urban pollution levels. Smart traffic signal optimization and ensemble learning-based flow prediction models have been shown to significantly cut idle times and fuel consumption, thereby reducing the environmental footprint of road transportation [38], [39]. This makes optimization a vital tool not only for improving mobility but also for advancing climate action goals, particularly in regions such as Nigeria where climate-related vulnerabilities already affect road infrastructure reliability [40].

Optimization enhances economic and social efficiency by ensuring reliable connectivity within and between cities. In logistics and supply chain management, optimized traffic systems reduce delays in the movement of goods and services, leading to lower operational costs and improved customer satisfaction [41]. Likewise, vehicular communication networks and connected autonomous vehicles (CAVs) rely on optimized data offloading and routing mechanisms to function effectively in real-time [42]. These benefits highlight that traffic optimization is not merely a technical challenge but a socio-economic necessity that underpins sustainable urban development, economic resilience, and improved quality of life.

2.4 Dynamic Route Guidance in Traffic Systems

Dynamic Route Guidance Systems (DRGS) are intelligent frameworks designed to provide real-time navigation support for vehicles by suggesting the most efficient routes based on prevailing traffic conditions. Unlike static navigation systems that rely on predefined maps, DRGS dynamically integrates live data streams such as congestion levels, accident reports, weather conditions, and road closures to guide drivers toward optimal paths [43]. This makes them especially relevant in urban areas where traffic conditions change rapidly and require adaptive decision-making.

A core component of DRGS is their reliance on algorithms for shortest and optimal path determination. For instance, modifications of classical algorithms such as Dijkstra's have been developed to allow dynamic weight adjustments based on traffic density and road constraints [44]. By embedding these adaptive routing algorithms, route guidance systems can continuously update the suggested paths as conditions change, reducing travel times and minimizing unnecessary congestion. Such adaptability ensures that road users benefit from more efficient journeys, while overall network performance improves.

Integration with Artificial Intelligence (AI) and predictive modeling has further advanced the efficiency of DRGS. AI-driven approaches enable systems to not only react to current conditions but also anticipate congestion patterns, thereby recommending preventive rerouting strategies [45]. For example, machine learning techniques analyze historical traffic data alongside real-time input to forecast future congestion hotspots, which can then inform dynamic route allocation. This predictive capability makes DRGS indispensable in smart city infrastructures.

Moreover, DRGS play a critical role in urban congestion management by distributing vehicular flows across the available road network. In urban centers like Enugu, the analysis of route-way dynamics reveals how uneven traffic distribution often worsens bottlenecks [46]. By guiding vehicles along alternative but less congested routes, DRGS help balance traffic loads, thus enhancing travel efficiency across entire networks rather than focusing solely on individual driver benefit. This network-wide optimization is crucial for reducing systemic congestion in rapidly growing cities.

The functionality of DRGS is also being enriched by geospatial analysis and digital mapping technologies. Geolocation data, when combined with adaptive signal control, can significantly improve route guidance accuracy[47]. Additionally, spatial distribution studies, such as those examining jetty and road access in Port Harcourt, highlight the importance of accessibility analysis in designing effective route guidance strategies [48]. By embedding such geospatial intelligence, DRGS can deliver more context-aware and location-specific guidance.

Finally, the application of DRGS extends to logistics, emergency services, and cost-sensitive routing problems. For example, traveling salesman problem (TSP)-based models have been employed in mail delivery systems to minimize cost and improve efficiency through optimized routing [49]. Similarly, in emergency response scenarios, DRGS can ensure that ambulances and fire services are directed through the fastest and most reliable routes, potentially saving lives. This versatility underscores the transformative potential of DRGS across multiple domains, from urban commuting to large-scale transportation networks.

Beyond personal and commercial vehicle use, integration with hybrid and electric vehicle systems is another area where DRGS have shown significant potential. By combining multistate vehicle-traffic information with energy management strategies, these systems can optimize routes not only for time but also for fuel or battery efficiency [50]. This capability is critical for reducing emissions in urban areas and promoting sustainable transportation practices. As cities push toward greener mobility, DRGS provide the intelligence needed to balance environmental concerns with mobility demands.

Another important dimension of DRGS is their connection to infrastructure optimization. Research on smart pavement design and boundary layer modeling shows that intelligent infrastructure can be embedded with sensors to provide continuous updates on road conditions, surface efficiency, and energy transfer effects [51]. Such data can be fed into DRGS, enriching their guidance accuracy and enabling more proactive rerouting. When integrated with pervious concrete and road material innovations that enhance drainage and durability, DRGS can recommend routes that are both faster and safer, especially during adverse weather conditions [52].

In view of these, DRGS contribute to broader transportation network resilience by supporting adaptive responses during disruptions such as accidents, road repairs, or natural disasters. For example, RoPax ferry designs in Nigeria's coastal waters emphasize pliability and adaptability to changing travel demands [53]. Applying a similar principle, DRGS provide a flexible routing framework that keeps urban systems functional under stress, ensuring that people and goods continue moving efficiently even in the face of disruptions. This adaptive quality highlights DRGS as a cornerstone of future smart mobility ecosystems.

2.5 Shortest Path Algorithms in Vehicular Traffic Optimization

The shortest path problem forms the foundation of vehicular traffic optimization, as it directly influences how vehicles navigate through complex and dynamic road networks. In urban mobility, determining the most efficient route is not merely about minimizing physical distance but also about accounting for real-time conditions such as congestion, traffic signals, accidents, and road closures. Over the years, researchers have developed a variety of algorithms designed to compute optimal or near-optimal routes that balance speed, accuracy, and scalability. These algorithms, ranging from classical approaches like Dijkstra's and Bellman-Ford to more advanced heuristic and AI-driven models, play a pivotal role in enhancing travel efficiency and reducing delays in transportation systems.

This section provides an overview of shortest path algorithms as they apply to vehicular traffic optimization. It highlights the importance of shortest path computation in traffic routing, the criteria used to evaluate algorithmic performance, and the practical considerations that determine their applicability in real-world contexts. By examining both theoretical principles and applied strategies, this section establishes the groundwork for comparing classical approaches with modern techniques, ultimately identifying gaps that motivate the need for improved algorithms in intelligent transportation systems.

The shortest path problem is central to the efficiency of vehicular traffic systems, as it determines the ability of vehicles to navigate urban networks with minimal travel time and cost. By identifying optimal routes, shortest path algorithms help reduce congestion, fuel consumption, and overall journey delays, which are particularly important in rapidly urbanizing environments like Abuja, Nigeria [54]. In such cities, poorly optimized routes not only increase operational inefficiencies but also contribute to elevated carbon emissions and driver stress. Thus, shortest path solutions form the computational backbone of sustainable and effective traffic management.

2.6 Classical Shortest Path Algorithms

A. Dijkstra's Algorithm

Dijkstra's algorithm is a foundational pathfinding method used to identify the shortest route between nodes in a weighted graph, making it highly relevant for traffic optimization where roads are modeled as edges with weights representing distance, time, or congestion. Its deterministic process guarantees accurate and reliable shortest paths, which is valuable for applications such as urban navigation, bus route planning, and intelligent traffic management [55],[56]. The algorithm has also been adapted across domains, from production planning to industrial automation, demonstrating its versatility [57], [58]. However, despite these strengths, Dijkstra's algorithm faces scalability and real-time adaptation challenges, as it requires significant computational resources and frequent recalculations to account for dynamic traffic conditions like accidents or sudden congestion [59]. These limitations have spurred modifications and hybrid approaches aimed at improving its efficiency and responsiveness in modern intelligent transportation systems.

B. A (A-Star) Algorithm

The A* (A-Star) algorithm is a heuristic-based pathfinding algorithm that extends Dijkstra's algorithm by incorporating a cost function, f(n)=g(n)+h(n)f(n) = g(n) + h(n), where g(n)g(n) is the actual cost from the start node to the current node, and h(n)h(n) is a heuristic estimate of the cost from the current node to the destination. This combination enables A* to find the shortest path more efficiently by prioritizing nodes that appear closer to the goal [60]. In traffic systems, A* has been widely applied to vehicle navigation, autonomous driving, and urban traffic routing due to its ability to balance accuracy and efficiency [61]. By integrating heuristic functions such as estimated travel time or congestion levels, A* adapts to real-world traffic networks, allowing vehicles to dynamically select routes that minimize delays and improve mobility.

C. Bellman-Ford Algorithm

The Bellman-Ford algorithm is a classical shortest pathfinding method that computes the minimum cost paths from a single source to all other nodes in a weighted graph, with the unique ability to handle negative edge weights, unlike Dijkstra's algorithm. Its mechanism is based on iterative relaxation of all edges up to (V−1) times, ensuring accurate results even in variable or uncertain environments such as traffic systems where costs may fluctuate due to toll adjustments, congestion pricing, or incomplete information [62], [63]. The algorithm's flexibility makes it advantageous in modeling complex transport networks, and it has been extended to fuzzy, probabilistic, and symbolic contexts to enhance decision-making in uncertain conditions [63]. However, its major limitation lies in its higher time complexity of O(V×E), which reduces efficiency in large-scale, real-time vehicular networks, where rapid recalculations are critical for dynamic routing [66]. In comparing the three algorithms in focus, we present them side by side in Table 1 as seen below;

Table 1: *Strengths and weaknesses of the three algorithms*

Algorithm	Strengths	Weaknesses
Dijkstra	- Efficient in static networks O((V+E)log V)) - Moderate memory usage - Guarantees optimal solution	- Poor adaptability to dynamic traffic - Requires recomputation for real-time updates
A*	- Faster with good heuristics - Effective for real-time rerouting - Balances accuracy and efficiency	- Higher memory usage -Performance depends heavily on heuristic quality
Bellman-Ford	- Handles negative edge weights -Works well in variable/uncertain environments - Simple implementation	- High time complexity O(VE)) -Slow convergence in large-scale real-time traffic systems

Review of different works done by other authors on optimization and dynamic traffic systems have been done, and we present their respective research gaps in Table 2 as below;

2.7 Gaps in the Current System and the Proposed Solution

Traditional shortest path algorithms such as Dijkstra, A*, and Bellman-Ford have been widely used in traffic optimization systems; however, they often operate with fixed road weights or distances, assuming static traffic conditions. This limitation prevents them from responding effectively to sudden road events such as congestion, accidents, or blockages. Among these algorithms, Dijkstra's Algorithm was chosen for this study because it is computationally efficient, consuming less memory and processing time compared to others. Nonetheless, its conventional form lacks the dynamism needed to reflect changing road conditions in real time.

To address this gap, the current work modifies the traditional Dijkstra's Algorithm by incorporating directional and cost adaptability into its computation process. The modified version allows the system to dynamically adjust when congestion or road incidents occur by reversing direction, rerouting vehicles, and reassigning road weights to reflect updated traffic conditions, which is something the existing system could not achieve [63]. This ensures that when a route becomes congested, the system automatically evaluates available alternatives and redirects users to a faster, less congested path. The integration of this dynamic behavior enables real-time rerouting, minimizes travel time, and prevents vehicles from remaining trapped in long queues. The modified algorithm transforms Dijkstra's static model into a flexible, intelligent, and responsive pathfinding mechanism, suitable for real-world traffic management in Calabar Metropolis.

3.3 METHODOLOGY

The methodology adopted for this research involves the design and implementation of a web-based intelligent traffic navigation system that leverages real-time data to provide users with optimal routes within Calabar Local Government Area, Cross River State. The system follows a layered architectural design that ensures seamless integration of data collection, processing, decision-making, and user interaction. The goal is to deliver a responsive, memory-efficient, and adaptive routing system that dynamically adjusts to traffic variations and road events.

3.1 Data Ingestion Layer

This is the foundational layer that connects to various dynamic traffic data collected from the user while on the go to the system. It continuously receives updates on road conditions including starting point, destination, incident(congestion) alerts, and directional flow changes. These inputs form the raw information required for the dynamic routing process, ensuring that the system operates based on current traffic realities rather than static assumptions. The conceptual system architecture of this work is presented in figure 2(a).

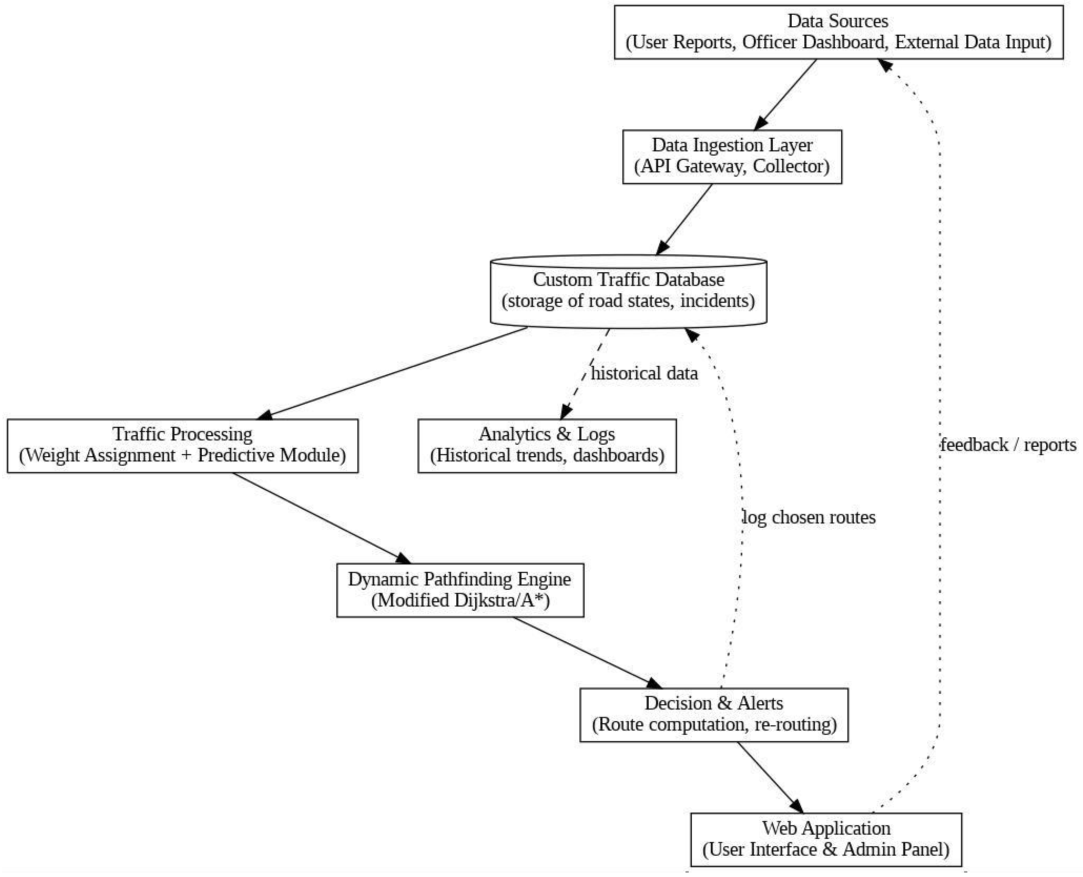

Figure 2 (a): *Conceptual System Architecture*

3.2 Weight Assignment Module

At this stage, the collected traffic data is transformed into graph edge weights. Roads are represented as edges in a weighted graph, while intersections or junctions act as nodes. Each road segment's weight reflects its current traffic state — low weights for free-flowing roads and higher weights for congested or obstructed routes. This transformation enables the conversion of real-world traffic conditions into a structured, computational format, forming the foundation for dynamic route optimization.

3.3 Dynamic Pathfinding Engine - Modified Dijkstra's Algorithm

The core computational aspect of the system resides here. A modified version of Dijkstra's Algorithm is employed due to its low memory consumption and fast processing speed, which make it suitable for large-scale, real-time navigation. However, while the traditional Dijkstra's algorithm assumes static road weights, the proposed system introduces dynamism to reflect real-time traffic fluctuations. In the modified version, road weights and directions are updated automatically whenever congestion, accidents, or road closures occur. Instead of recomputing the entire graph when conditions change, the algorithm selectively updates only the affected nodes and edges, significantly reducing computation time while maintaining high accuracy. This adaptive capability enables the system to reverse direction, recalculate new paths, and reassign costs dynamically, ensuring that users are always guided along the most efficient route.

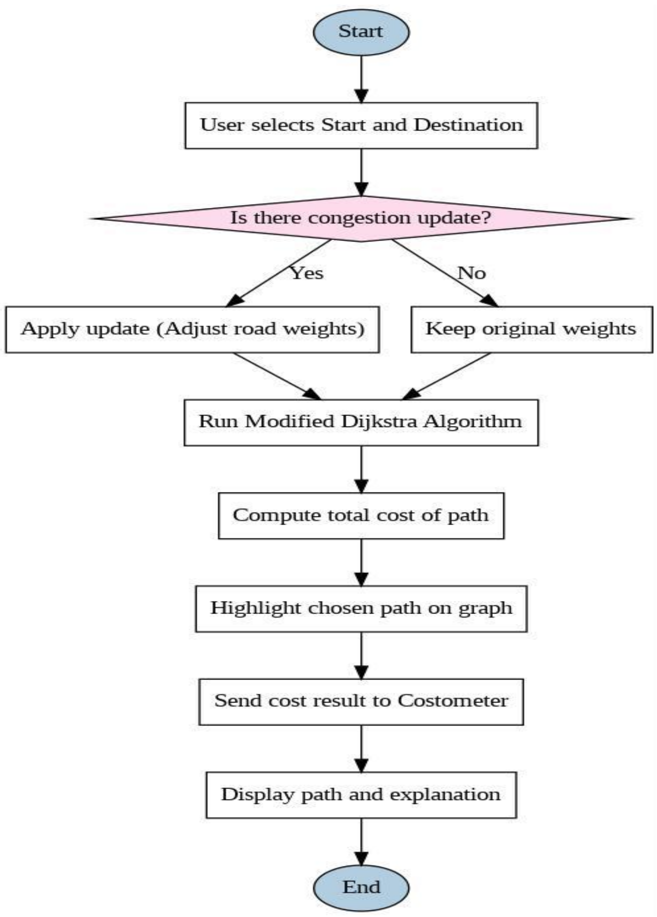

Figure 2(b): Flowchart of the System

3.4 Decision Layer

The Decision Layer interprets outputs from the modified pathfinding engine and transforms them into user guidance actions. Once the optimal route is computed, it is relayed to the user in real time. If traffic conditions deteriorate or unexpected events arise while the driver is en route, the Decision Layer automatically triggers a rerouting process to compute an alternative path, ensuring continuous and uninterrupted navigation.

3.5 User Interface Layer

The web-based user interface serves as the interactive platform where users visualize their routes, traffic updates, and rerouting notifications. Designed with a modern UI/UX approach, it features a clean map display, color-coded traffic indicators, and easy-to-read route suggestions. Users can access the system seamlessly via desktops, tablets, or smartphones, enabling widespread usability. The interactive design ensures intuitive navigation, minimal cognitive load, and real-time responsiveness to system updates.

4. RESULTS AND DISCUSSION

This section presents results obtained from the implemented system and interprets them in the context of the research objectives. We begin by summarizing how the system was developed (start/destination pairs, officer updates, congestion toggles and reversal/diversion choices) and what kinds of outputs were collected: route lists, path visualizations, the costometer gauge values, and diversion/deadlock decisions. The input data used are the in-memory network and manually injected updates representing real-time officer reports. The results are organized to showcase (1) baseline routing behavior under no congestion, (2) behavior under single-edge congestion updates including the "block original path" policy, (3) the diversion logic when reversing is not possible, and (4) the costometer outputs that represent the computed travel cost dynamically. Each panel brings a concise textual result with a visualization and a short interpretive discussion so findings are reproducible and traceable to the algorithmic decisions made in implementation. In figure 3 below, we present the full system interface.

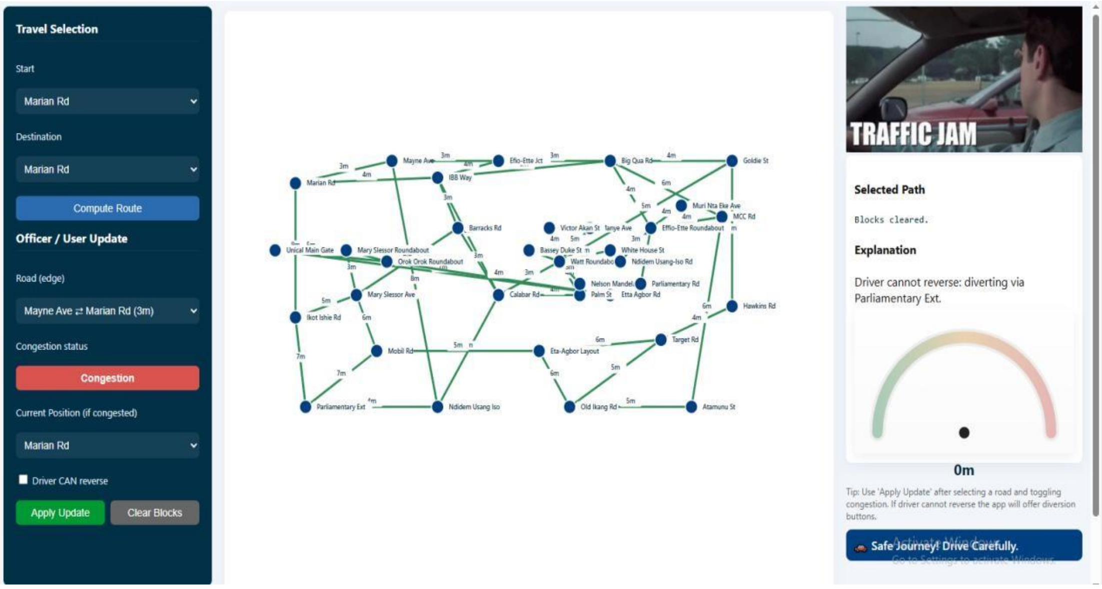

Figure 3: *Full system UI with network and costometer visible*

In Figure 3, the interface is composed of three functional regions: the left sidebar (controls: start, destination, current, choose edge, toggle congestion, reversal checkbox, apply/clear), the central SVG map (network graph with edges, nodes, and inline weight labels), and the right panel that contains the selected path, explanation text and the costometer gauge. The costometer visually maps the computed route cost onto a semi-circular gauge for quick perception by users, and shows the numeric total below the gauge. Graphical affordances include colored edges representing relative congestion (weight/base ratio), special "NO ROAD" styling for blocked edges, highlighted nodes/edges for original and updated routes (cyan and orange respectively), and background white rectangles behind edge labels to maintain readability when edges and labels are close. The central map is interactive only in the sense that control changes update visualization — nodes are not dragged but are positioned to minimize overlaps. This UI layout allows an officer or driver to (a) simulate congestion events, (b) evaluate whether reversing is possible, (c) see available diversion options (buttons appear when reversing is disallowed and local neighboring roads exist), and (d) immediately see the costometer update reflecting the recalculated travel cost. The diagram in Figure 4 illustrates the logical structure used in the system, showing how nodes, edges, and update actions interact to support real-time path recalculation.

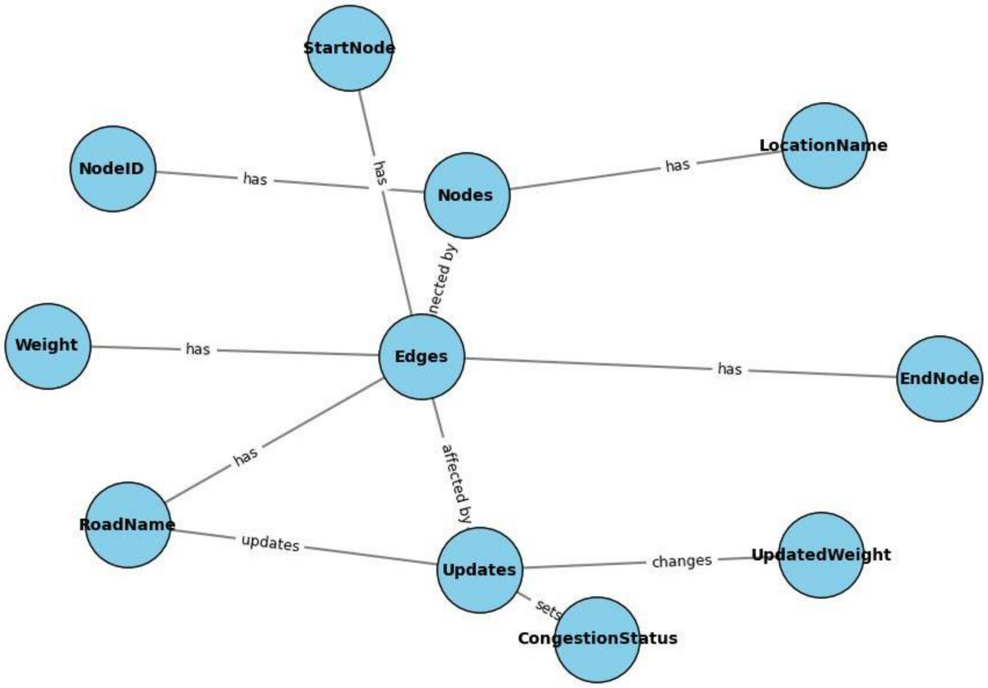

Figure 4: *Database Structure of the Proposed System*

The database structure for the developed system as seen in Figure 4, is implemented directly within the program using in-memory data representations in JavaScript. This structure captures the core elements required for traffic optimization, namely nodes (intersections or locations), roads (edges between nodes), and the weights that represent travel costs. Each road is associated with a numerical weight that reflects travel conditions, which can change when congestion is reported by users or officers. This simple but effective structure ensures that the modified Dijkstra's algorithm always processes the most recent conditions in order to compute the optimal travel path.

The structure consists of three main components. First, the nodes are represented as key locations within Calabar, such as Watt Market, Mary Slessor Avenue, or Marian Road. These nodes form the endpoints of the network graph. Second, the edges connect these nodes and include attributes such as the road name and the travel cost. Initially, costs are low to represent free-flow conditions, but when congestion is indicated, the weights are dynamically increased to reflect delays. Finally, the update records are applied directly within the program, where congestion status for a selected road is changed by the user through the interface. This triggers an immediate recalculation of the shortest path, with both the original and updated routes displayed alongside their respective costs. For the database features and type, we present their descriptions in Table 3.

Table 3: *Data Features (Nodes) and Type*

Field Name	**Data Type**	**Description**
NodeID	Integer	Unique identifier for each road junction
LocationName	Varchar(50)	Name of the junction or road intersection

The database for the proposed traffic optimization system was designed to be simple yet efficient, focusing on the minimum entities required to capture the road network, traffic conditions, and dynamic updates. Three major tables were implemented: Nodes Table as seen in Table 3, Edges Table presented in Table 4, and Updates Table as seen in Table 5. These tables form the foundation for storing graph-related information used by the modified Dijkstra's algorithm to compute optimal routes in real time. The Nodes Table stores all points of interest (intersections or junctions) in the road network. Each node has a unique identifier and a location name that corresponds to real road junctions in Calabar, Cross River State.

Table 4: Edges Table

Field Name	Data Type	Description
EdgeID	Integer	Unique identifier for each road segment
StartNode	Integer	ID of the starting junction (foreign key to Nodes)
EndNode	Integer	ID of the destination junction (foreign key to Nodes)
RoadName	Varchar(50)	Name of the road segment
Weight	Float	Travel cost or time assigned to the road

The Edges Table in Table 4 stores all connections between nodes, representing actual roads. For each edge, the start node, end node, road name, and weight (travel cost or time) are defined.

Table 5: Updates table

Field Name	Data Type	Description
UpdateID	**Integer**	**Unique identifier for each update**
RoadName	**Varchar(50)**	**Name of the road being updated**
CongestionStatus	**Boolean**	**Indicates whether congestion is reported (Yes/No)**
UpdatedWeight	**Float**	**New travel cost assigned to the road after update**

The Updates Table presented in Table 5 records the latest traffic conditions as reported by officers or users, where a congestion status can be toggled, and an updated weight value is applied to the corresponding road. This design ensures that the graph structure dynamically adapts to changing road conditions while preserving consistency in path computations.

4.1 Pathfinding Without Congestion (Baseline Results)

Under baseline conditions (no congestion toggles and blockedEdgeIds empty), the modified Dijkstra algorithm reduces to standard Dijkstra: the cost of a path (P = (v_0, v_1, ..., v_k)) is computed as

$$(\mathrm{cost}(P) = \sum_{i=0}^{k-1} w(v_i, v_{i+1})) \ldots\ldots\ldots\ldots [1]$$

with each (w) initialized to the edge base. The algorithm selects a path that minimizes this additive cost. For several test start/destination pairs we show the route string, the total cost (displayed textually), and the costometer reading. Experimental observations: baseline routes tend to follow the main arteries (e.g., Calabar Rd → Efio-Ette → Big Qua → MCC). The costometer mapping uses a precomputed gauge scale gaugeMax derived by scanning max baseline path values across node pairs (with a safety margin) so the gauge has sensible dynamic range. Baseline costs serve as the reference for subsequent congestion experiments. Report these baseline runs as Table rows (start, dest, path, costometer value) and include one screenshot with the highlighted path and gauge at the reported cost. This helps quantify the incremental cost produced by later congestion updates.

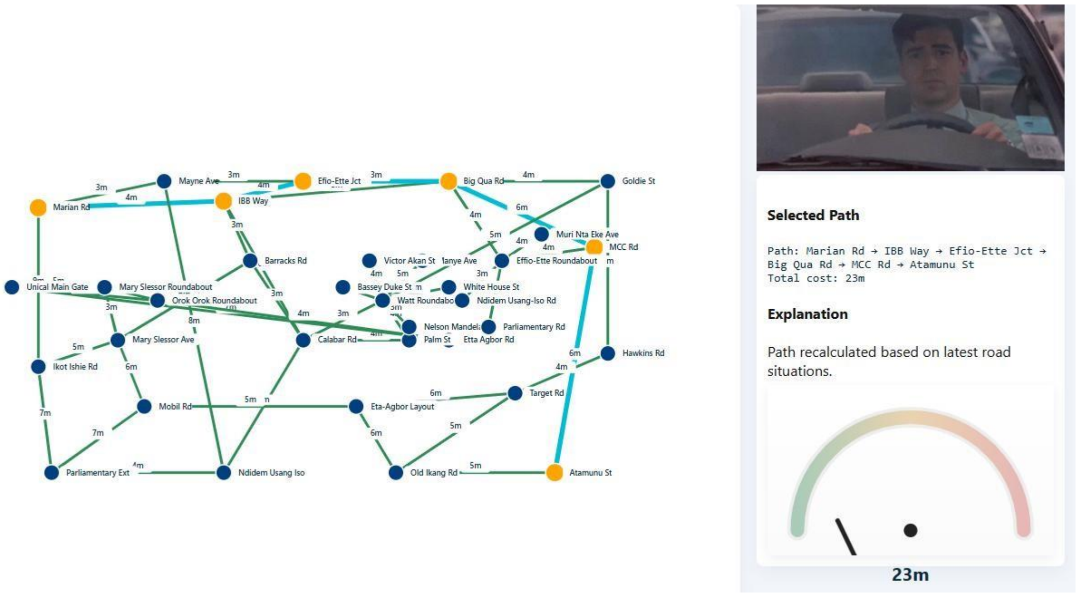

Figure 5: Example baseline path and costometer reading

4.2 Pathfinding With Congestion Updates

When a congestion event is applied to a selected edge id, its effective weight is updated (in the experiments, weight := base * 3). Additionally, when the "block original path" policy is in force the implementation adds all edges that comprised the original route to blockedEdgeIds. The modified shortest-path computation then proceeds in the same additive manner but with the set of available edges reduced by blocked edges. Formally, the relaxation step is unchanged:

$$(\mathrm{dist}[v] \leftarrow \min(\mathrm{dist}[v], \mathrm{dist}[u] + w(u,v)))\ldots\ldots[2]$$

but neighbors linked by blocked id are skipped entirely. The practical outcome is either (a) a detour route that circumvents the blocked sequence, (b) a longer route that still uses some remaining unblocked edges, or (c) no path found when the network is partitioned by blocks. The system records the original route (cyan) and the updated route (orange) and displays both to help an officer see the effect of the update, while the costometer shows the new remaining (or combined) cost. Include an example where congestion is placed near Watt Roundabout and compare the original vs updated path and costs. Empirical output is illustrated with screenshots and a short quantitative table showing original cost, updated remaining cost (from current or start as relevant), and the combined cost when the driver has traversed part of the original route.

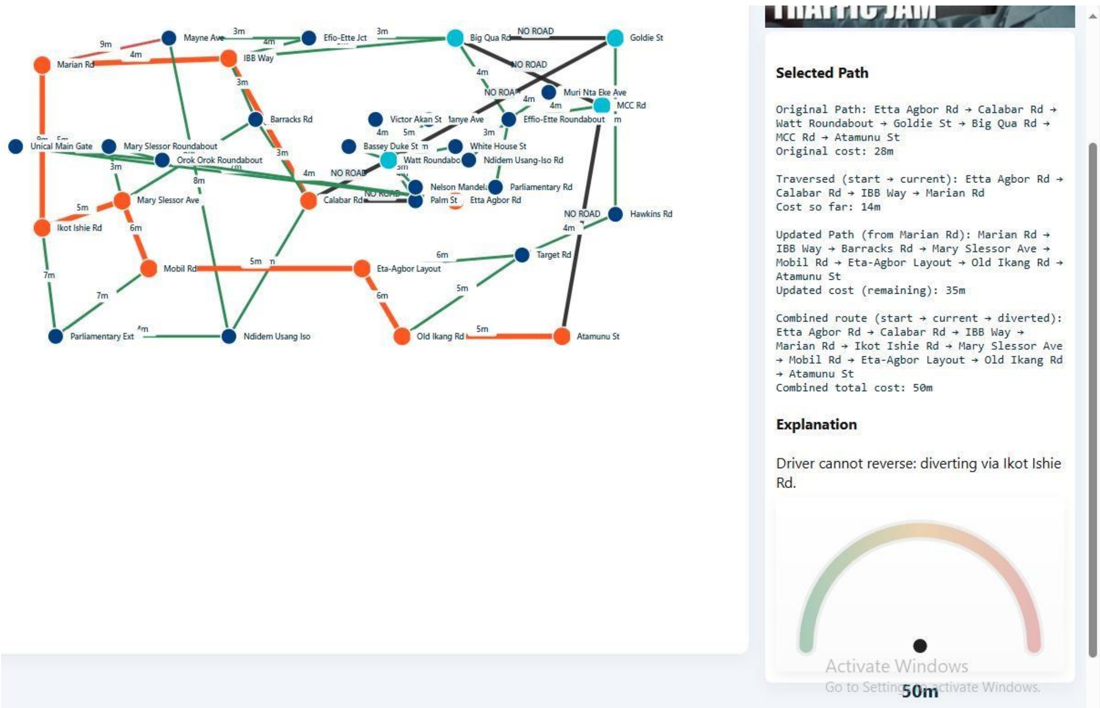

Figure 6: Original and updated paths with blocked edges shown as NO ROAD

4.3 Costometer (Dynamic Travel Cost Display)

The costometer is a semi-circular gauge that maps computed path cost to an angle on the dial. We scale the needle by gaugeMax, which is computed by scanning the (temporarily unblocked) pairwise shortest path max cost across the graph and applying a safety multiplier (1.2). The visual mapping is linear:

$$(\theta = -110^\circ + 220^\circ \cdot \frac{\text{cost}}{\text{gaugeMax}}) \ldots\ldots\ldots\ldots [3]$$

The numeric display below the gauge shows cost in the same human units used for edge weights (m/min). If no path exists the numeric display shows infinity (∞) and the needle is clamped to the max. One important behavior: the costometer displays either (a) the total cost of the single computed route (compute button), (b) the remaining cost from the current position after an update, or (c) the combined cost (cost(start→current) + cost(current→dest)) when the driver has traversed a portion and reversing is allowed. The gauge helps the user quickly assess the magnitude of penalty introduced by congestion compared to baseline. For reporting, present several snapshots showing the gauge at baseline, after moderate congestion, and after heavy congestion (including deadlock/infinite reading). A small table of numeric values should accompany the images for precise comparison.

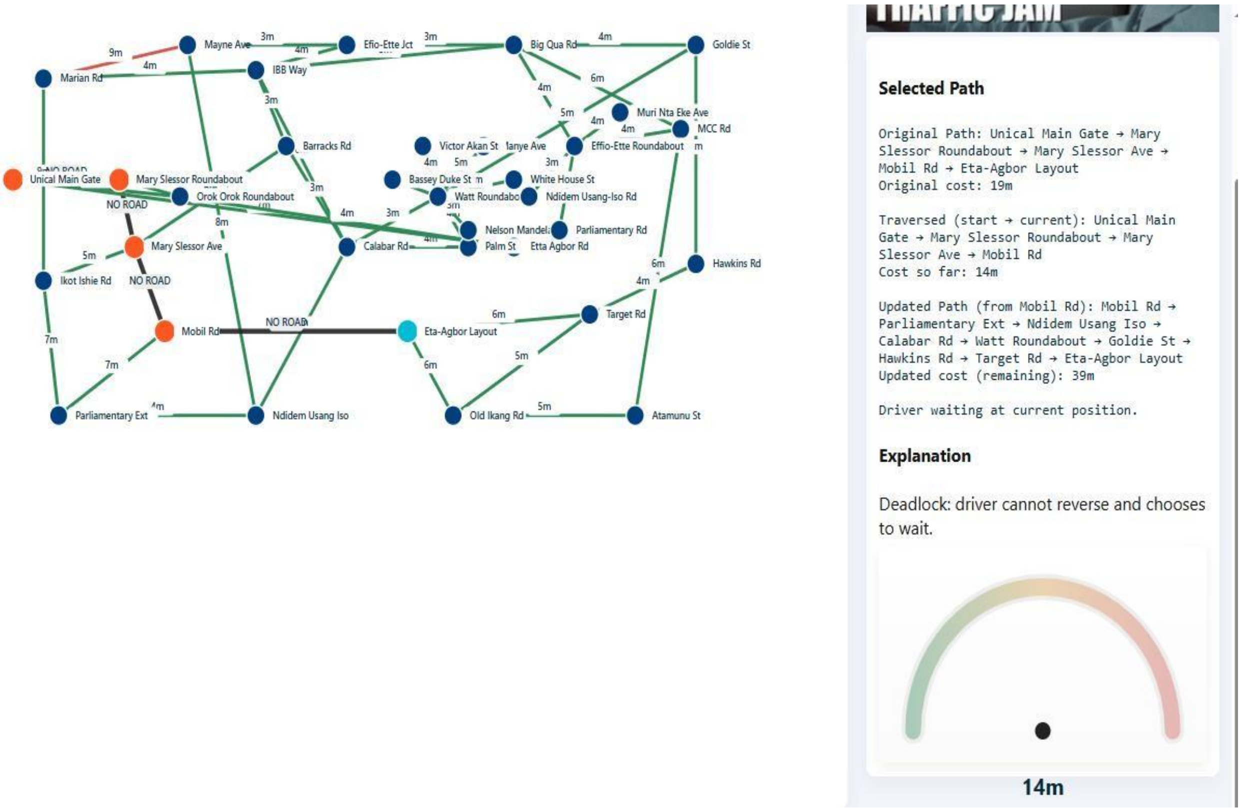

Figure 7: Costometer in three states: baseline, congested, deadlock

4.4 User Road Update Functionality

Users update via the left-side controls; a user selects an edge, toggles the congestion button, marks the current position (if appropriate), and clicks Apply. The system then (1) records the changed edge weight, (2) computes the original route and the traversed portion (start→current), (3) if congestion is active adds original-route edges to blockedEdgeIds, and (4) computes new routes respecting blocked edges. The UI responds with updated path displays, the costometer, and diversion options if reversing is disallowed.

This flow was tested across multiple update cases, including isolated single-edge congestion, congestion on a sequence of edges, and updates that effectively partition the graph. The officer-level control allowed us to reproduce scenarios where driver decisions matter: whether to reverse, to divert, or to wait. The “Deadlock — Wait” option simulates the real-world situation where neither forward nor backward movement is possible.

Document these interactions as a sequence of short-case narratives in the thesis: starting state (UI screenshot), officer update action, resulting UI state and costometer value, and the final decision chosen (divert via X or wait).

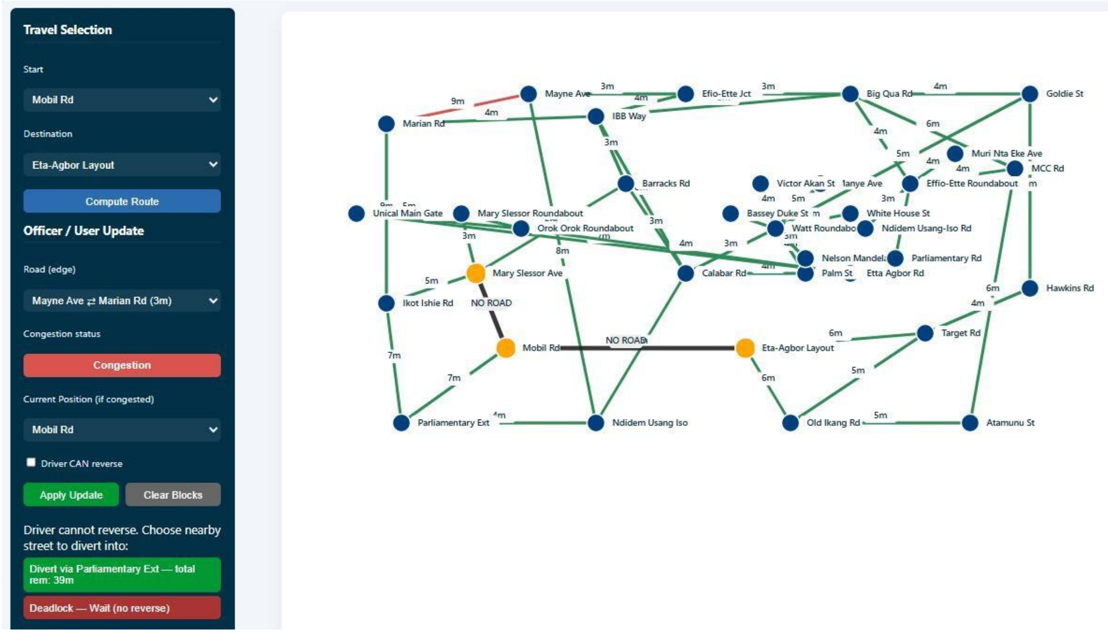

Figure 8: Example of officer update interaction and resultant UI states

4.5 Comparing Old Path vs New Path after Congestion Update

We present side-by-side comparisons of original (pre-update) and updated (post-update) routes for representative origin–destination pairs. Quantitatively, report the original total cost (C_o), the new remaining cost (C_r), and the percentage increase

$$(\Delta = 100 \cdot \frac{C_r - C_o}{C_o}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots [4]$$

where applicable. When the combined route is used (start→current→new-route), report the combined cost which is expressed as seen below;

$$\left(C_c = C_{start \to current} + C_{current \to dest}\right) \ldots\ldots\ldots\ldots [5]$$

The example in Figure 9 shows that local edge congestion can cause modest detours (10–40% cost increase) or significant rerouting when the original route is blocked entirely. In some experiments blocking the original route forced the algorithm to route through longer branches of the network (e.g., detouring via Big Qua → IBB Way instead of direct Calabar Rd), with corresponding changes in edge-level colorings that make the magnitude of change visually apparent.

Visual figures (overlay maps and path-highlighted snapshots) should accompany each quantitative example, and a small table listing (C_o, C_r, C_c) helps readers quickly understand trade-offs between waiting (if allowed) and taking a diversion.

Figure 9: Side-by-side before/after route comparison with table of costs

5. CONCLUSION

The study successfully developed and implemented a system that applies a modified Dijkstra's algorithm for shortest path computation with congestion handling. The algorithm proved capable of dynamically recalculating new routes when blockages occurred, while ensuring that previously traversed costs were not ignored. This outcome validates the robustness and practical relevance of the modification.

The integration of dynamic cost tracking through the costometer and the visualization of updated paths represent significant advancements over traditional pathfinding models. The study establishes a foundation for further research in intelligent routing systems and demonstrates how algorithmic improvements can contribute to solving real-world challenges in transportation and mobility planning.

REFERENCE

1. Abdussalam, A. F., Isa, Z., Babati, A., Baba, B. M., Suleiman, A., Abubakar, A. S., ... & Ibrahim, M. N. (2025). Multivariate bias correction of ERA5 climatic data for assessing climate-related road vulnerabilities in Nigeria. Discover Geoscience, 3(1), 38.
2. Chukwurah, G. O., Okeke, F. O., Isimah, M. O., Nnaemeka-Okeke, R., Okonta, E. D., Awe, F. C., ... & Okeke, C. A. (2025). Analysis of Route-Way Dynamics in Urban Traffic Congestion of Enugu, Nigeria. Future Transportation, 5(2), 71.
3. Ahmed, M., Mirza, M. A., Raza, S., Ahmad, H., Xu, F., Khan, W. U., ... & Han, Z. (2023). Vehicular communication network enabled CAV data offloading: A review. IEEE Transactions on Intelligent Transportation Systems, 24(8), 7869–7897.
4. Alkarim, A. S., Al-Ghamdi, A. S. A. M., & Ragab, M. (2024). Ensemble learning-based algorithms for traffic flow prediction in smart traffic systems. Engineering, Technology & Applied Science Research, 14(2), 13090–13094.
5. Alam, M. G. R., Suma, T. M., Uddin, S. M., Siam, M. B. A. K., Mahbub, M. S. B., Hassan, M. M., & Fortino, G. (2021). Queueing theory based vehicular traffic management system through Jackson network model and optimization. IEEE Access, 9, 136018–136031.
6. Alshammrei, S., Boubaker, S., & Kolsi, L. (2022). Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. Comput. Mater. Contin, 72(3), 5939–5954.
7. Behún, M., Knežo, D., Cehlár, M., Knapčíková, L., & Behúnová, A. (2022). Recent application of Dijkstra's algorithm in the process of production planning. Applied Sciences, 12(14), 7088.
8. Adedeji, J. A., Feikie, X. E., & Dijanic, D. (2024). Driver-to-driver communication skills: Contributing factors to traffic accidents on South African roads?. Sustainability, 16(11), 4833.
9. Alshayeb, S., Stevanovic, A., Mitrovic, N., & Espino, E. (2022). Traffic signal optimization to improve sustainability: A literature review. Energies, 15(22), 8452.
10. Dikshit, S., Atiq, A., Shahid, M., Dwivedi, V., & Thusu, A. (2023). The use of artificial intelligence to optimize the routing of vehicles and reduce traffic congestion in urban areas. EAI Endorsed Transactions on Energy Web, 10, 1–13.
11. Berhanu, Y., Alemayehu, E., & Schröder, D. (2023). Examining car accident prediction techniques and road traffic congestion: A comparative analysis of road safety and prevention of world challenges in low-income and high-income countries. Journal of Advanced Transportation, 2023(1), 6643412.
12. Titus, A. I., Okonwko, C. C., Egbe, J. G., & Onwukwe, C. E. (2025). Analysis of traffic congestion in a network of roads at the Calabar Effio-Ette intersection at the microscopic level. JCEA, 6(1), 10–22.
13. Akpado, K., Usoro, S., & Ezeani, N. (2021). An Optimised Traffic Control Scheme with Preference to Emergency Vehicles Leveraging RFID. Journal of Engineering Research and Reports, 20(10), 74–89.
14. Alaba, F. A., Oluwadare, A., Sani, U., Oriyomi, A. A., Lucy, A. O., & Najeem, O. (2024). Enabling sustainable transportation through IoT and AIoT innovations. In Artificial Intelligence of Things for Achieving Sustainable Development Goals (pp. 273–291). Cham: Springer Nature Switzerland.
15. Bhargava, A., Bhargava, D., Kumar, P. N., Sajja, G. S., & Ray, S. (2022). Industrial IoT and AI implementation in vehicular logistics and supply chain management for vehicle mediated transportation systems. International Journal of System Assurance Engineering and Management, 13(Suppl 1), 673–680.
16. Ekong, A., Ekong, B., & Edet, A. (2022). Supervised machine learning model for effective classification

of patients with COVID-19 symptoms based on Bayesian belief network. Researchers Journal of Science and Technology, 2(1), 27–33.

17. Ekong, B., Ekong, O., Silas, A., Edet, A. E., & William, B. (2023). Machine Learning Approach for Classification of Sickle Cell Anemia in Teenagers Based on Bayesian Network. Journal of Information Systems and Informatics, 5(4), 1793–1808.

18. Edet, A. E., & Ansa, G. O. (2023). Machine learning enabled system for intelligent classification of host-based intrusion severity. Global Journal of Engineering and Technology Advances, 16(03), 041–050.

19. Ebong, O., Edet, A., Uwah, A., & Udoetor, N. (2024). Comprehensive Impact Assessment of Intrusion Detection and Mitigation Strategies Using Support Vector Machine Classification. Research Journal of Pure Science and Technology, 7(2), 50–69.

20. Uwah, A., & Edet, A. (2024). Customized Web Application for Addressing Language Model Misalignment through Reinforcement Learning from Human Feedback. World Journal of Innovation and Modern Technology, 8(1), 62–71.

21. Edet, A., Ekong, B., & Attih, I. (2024). Machine Learning Enabled System for Health Impact Assessment of Soft Drink Consumption Using Ensemble Learning Technique. International Journal of Computer Science and Mathematical Theory, 10(1), 79–101.

22. Ekong, A., James, G., Ekpe, G., Edet, A., & Dominic, E. A. (2024). Model for the Classification of Bladder State Based on Bayesian Network. International Journal of Engineering and Artificial Intelligence, 5(2), 33–47.

23. Ekong, B., Edet, A., Udonna, U., Uwah, A., & Udoetor, N. (2024). Machine Learning Model for Adverse Drug Reaction Detection Based on Naive Bayes and XGBoost Algorithm. British Journal of Computer, Networking and Information Technology, 7(2), 97–114.

24. Edet, A., Udonna, U., Attih, I., & Uwah, A. (2024). Security Framework for Detection of Denial of Service (DoS) Attack on Virtual Private Networks for Efficient Data Transmission. Research Journal of Pure Science and Technology, 7(1), 71–81.

25. Udoetor, N., Ansa, G., Ekong, A., & Edet, A. (2024). Intelligent System for Detection of Copyright-Protected Data for Enhanced Data Security. Technology, 7(4), 58–80.

26. Edet, A., Silas, A., Ekaetor, E., Ebong, O., Isaac, E., & Udoetor, N. (2024). Data-Driven Framework for Classification and Management of Start-Up Risk for High Investment Returns. Advanced Journal of Science, Technology and Engineering, 4(2), 81–102.

27. Edet, A., Inyang, S., Umoren, I., & Etuk, U. E. (2024). Machine Learning Approach for Classification of Cyber Threat Actors in Web Region. Journal of Technology and Informatics (JoTI), 6(1), 70–77.

28. Edet, A., Godwin, H., Chukwuemeke, O., & Hanson, Y. (2025). Economic and cost-benefit analysis of diabetes treatment: correlating treatment cost with clinical outcomes using multi-regression approach. International Journal of General Scientific Research and Application, 1(1), 1–19.

29. Edet, A., Chukwuemeke, O., Godwin, H., & Eyo, E. (2025). Intelligent Framework for Classification of Influenza Severity for Respiratory Disease Management. Research Journal of Pure Science and Technology, 8(6), 29–46.

30. Ekong, A., James, G., Godwin Ekpe, I. J., Ekong, O., Ekong, B., Edet, A., & Okon, E. (2025). Support Vector Machine-Based Model for the Classification of Effective Antibiotic Combination for Pediatrics: A Comparative Analysis of Pre-COVID-19 and COVID-19 Era. European Journal of Science, Innovation and Technology, 5(3), 190–210.

31. Edet, A., Johnson, N., Godwin, H., Agashu, J. A., & Etim, H. (2025). Analytical framework for investigating privacy preserving techniques in IoTs ecosystem. International Journal of General Scientific Research and Application (IJGSRA), 1(4).

32. Fameso, F., Ndambuki, J., Kupolati, W., & Snyman, J. (2024). Boundary Layer Modelling of Flow Acceleration and Energy Transfer Effects in Smart Pavement Design. Science, Engineering and Technology, 4(2), 1–13.

33. Wang, J., Jiang, S., Qiu, Y., Zhang, Y., Ying, J., & Du, Y. (2021). Traffic signal optimization under connected-vehicle environment: An overview. Journal of Advanced Transportation, 2021(1), 3584569.

34. Wang, X., Jerome, Z., Wang, Z., Zhang, C., Shen, S., Kumar, V. V., ... & Liu, H. X. (2024). Traffic light optimization with low penetration rate vehicle

trajectory data. Nature Communications, 15(1), 1306.

35. Yan, Z., Kreidieh, A. R., Vinitsky, E., Bayen, A. M., & Wu, C. (2022). Unified automatic control of vehicular systems with reinforcement learning. IEEE Transactions on Automation Science and Engineering, 20(2), 789–804.

36. Olayode, I. O., Tartibu, L. K., Okwu, M. O., & Ukaegbu, U. F. (2021). Development of a hybrid artificial neural network-particle swarm optimization model for the modelling of traffic flow of vehicles at signalized road intersections. Applied Sciences, 11(18), 8387.

37. Yang, Y., Wang, W., Liu, L., Dev, K., & Qureshi, N. M. F. (2022). AoI optimization in the UAV-aided traffic monitoring network under attack: A Stackelberg game viewpoint. IEEE Transactions on Intelligent Transportation Systems, 24(1), 932–941.

38. Jurczenia, K., & Rak, J. (2022). A survey of vehicular network systems for road traffic management. IEEE Access, 10, 42365–42385.

39. Jabbar, L. S., Abass, E. I., & Hasan, S. D. (2023). A modification of shortest path algorithm according to adjustable weights based on Dijkstra algorithm. Engineering and Technology Journal, 41(2), 359–374.

40. Di Caprio, D., Ebrahimnejad, A., Alrezaamiri, H., & Santos-Arteaga, F. J. (2022). A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights. Alexandria Engineering Journal, 61(5), 3403–3415.

41. Wayahdi, M. R., Ginting, S. H. N., & Syahputra, D. (2021). Greedy, A-Star, and Dijkstra's algorithms in finding shortest path. International Journal of Advances in Data and Information Systems, 2(1), 45–52.

42. James, G. G., Oduobuk, E. J., Akpan, B. E., Judah, A. R., Ekah, U. J., Emeruwa, C., & Abraham, E. Smart Urban Traffic Management: Integrating GPS Geolocation and Adaptive Traffic Signal Control for Enhanced Efficiency and Safety. Available at SSRN 5093091.

43. Lawrence, H., & Eke, G. C. (2025). Geospatial Analysis of Jetty Distribution and Accessibility in Port Harcourt Metropolis, Rivers State, Nigeria. Journal of Spatial Information Sciences, 2(1), 144–158.

44. Brendan, N. G., & Eko, C. E. (2023). Travelling salesman's problem-based model application for mail routing and delivery with minimal cost. International Journal of Research and Innovation in Applied Science, 8(3), 113–129.

45. He, H., Wang, Y., Li, J., Dou, J., Lian, R., & Li, Y. (2021). An improved energy management strategy for hybrid electric vehicles integrating multistates of vehicle-traffic information. IEEE Transactions on Transportation Electrification, 7(3), 1161–1172.

46. Ewa, D. E., Ukpata, J. O., Otu, O. N., & Alaneme, G. U. (2023). Optimization of saw dust ash and quarry dust pervious concrete's compressive strength using Scheffe's simplex lattice method. Innovative Infrastructure Solutions, 8(1), 64.

47. Effiong Udo, A., & Daniel, T. (2022). Preliminary design analysis of RoPax for Nigeria coastal water pliability. Engineering, 14(10), 441–462.

48. Olonisakin, K., Murana, A., Abejide, O., & Olowosulu, A. (2024). Optimization of Bus Routes Network in Federal Capital City Abuja, Nigeria. FUOYE Journal of Engineering and Technology, 9(3).

49. Feijen, W., & Schäfer, G. (2021). Dijkstra's algorithm with predictions to solve the single-source many-targets shortest-path problem. arXiv preprint arXiv:2112.11927.

50. Mandloi, D., Arya, R., & Verma, A. K. (2021). Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3D environment. International Journal of System Assurance Engineering and Management, 12(5), 990–1000.

51. Rafai, A. N. A., Adzhar, N., & Jaini, N. I. (2022). A review on path planning and obstacle avoidance algorithms for autonomous mobile robots. Journal of Robotics, 2022(1), 2538220.

52. Lotfi, M., Osório, G. J., Javadi, M. S., Ashraf, A., Zahran, M., Samih, G., & Catalão, J. P. (2021). A Dijkstra-inspired graph algorithm for fully autonomous tasking in industrial applications. IEEE Transactions on Industry Applications, 57(5), 5448–5460.

53. Fujita, T., & Smarandache, F. (2024). Fundamental computational problems and algorithms for superhypergraphs. In Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization.

54. Tang, G., Tang, C., Claramunt, C., Hu, X., & Zhou, P. (2021). Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment. IEEE Access, 9, 59196–59210.

55. Lin, Z., Wu, K., Shen, R., Yu, X., & Huang, S. (2023). An efficient and accurate A-star algorithm for autonomous vehicle path planning. IEEE Transactions on Vehicular Technology, 73(6), 9003–9008.

56. Arnau, R., Calabuig, J. M., García-Raffi, L. M., Sánchez Pérez, E. A., & Sanjuan, S. (2024). A Bellman–Ford Algorithm for the Path-Length-Weighted Distance in Graphs. Mathematics, 12(16), 2590.

57. Parimala, M., Jafari, S., Riaz, M., & Aslam, M. (2021). Applying the Dijkstra algorithm to solve a linear Diophantine fuzzy environment. Symmetry, 13(9), 1616.

58. Kannan, V., & Appasamy, S. (2023). Employing the Bellman-Ford Algorithm with Score Functions to Address the Linear Diophantine Fuzzy Shortest Path Problem in Network Analysis. Mathematical Modelling of Engineering Problems, 10(5).

59. Kreuzer, M., Weber, A., & Knoll, A. (2025). A modified Bellman-Ford Algorithm for application in symbolic optimal control and plan and goal recognition. arXiv preprint arXiv:2501.17645.

60. Safari, M., Oortwijn, W., & Huisman, M. (2021). Automated verification of the parallel Bellman–Ford algorithm. In International Static Analysis Symposium (pp. 346–358). Springer.

61. Udoeka, I. J., Agana, M. A., Ofem, O. A., & Udo, I. J. (2024). Deep Learning Framework for Predicting Alternative Routes for Information Management in Intra-city Road Traffic Network.

62. Ofem, O. A., Edim, E. A., & Ele, S. I. (2019). Shortest Path Determination in a Wireless Packet Switch Network System in University of Calabar Using a Modified Dijkstra's Algorithm.