This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Emergency Routing Protocol for Intelligent Transportation Systems Using IoT and Generative Artificial Intelligence

Shijun Song and Min Fan

***Abstract*— Urban populations are always on the increase, increasing the number of vehicles, leading to severe congestion, longer travel times, and substantial emergency vehicle response time delays. This paper presents a routing protocol for Intelligent Transportation Systems (ITS), incorporating a Belief-Desire-Intention (BDI) model and generative Artificial Intelligence (AI) techniques. The proposed system utilizes BDI-based generative AI models, which rely on predefined logic and rules rather than machine learning to make real-time routing decisions. These decisions are based on data from vehicle-to-vehicle (V2V) communications, roadside units (RSUs) and IoT sensors, which integrate traffic density, congestion levels, collision avoidance and hazard detection. The system aims to optimize the accuracy of route selection, energy consumption, and communication latency. The simulation results, implemented using NS3 for real-world traffic scenarios, show improvements in route selection accuracy, collision avoidance, and energy efficiency compared to traditional routing methods. Specifically, the proposed system achieved a route selection accuracy of 95%, a collision avoidance rate of 95%, and reduced the communication latency to 105 ms, outperforming the other two methods. Furthermore, energy consumption was minimized and reduced to 85 J per route. These results highlight the potential of BDI-based routing with generative AI to improve ITS performance, particularly in real-time traffic management.**

***Index Terms*— BDI-based decision making, real-time routing, intelligent transportation systems, generative AI, NS3 simulation.**

I. Background

TO ENABLE better traffic management, mobility enhancement, and on-time emergency management, modern cities depend more and more on the so-called ITS. ITS utilizes connected emerging technologies like the Internet of Things (IoT), AI, blockchain, and digital twins to ensure efficient transportation operations. Despite this, current ITS implementations frequently under-perform in the realm of emergency response management, typically relying on fixed, rule-based protocols and having limited capacity for real-time data processing [1], [2].

Emergencies such as accidents, fires, or medical crises require transportation systems to respond quickly by prioritizing emergency vehicle routes. Unfortunately, many ITS deployments are unable to dynamically adjust to real-time traffic situations, causing critical delays that can result in life-threatening consequences [3]. Conventional routing protocols are static and cannot adapt promptly to unforeseen conditions such as road blockages, sudden weather changes, or traffic congestion. Furthermore, centralized ITS architectures often suffer from communication bottlenecks and system breakdowns due to reliance on a single-point data processing system [4].

Technological developments include digital twins and blockchain technologies and have the potential to address some of the limitations of intelligent transportation systems. Digital twins allow for real-time simulation and decision-making as they integrate different data streams continuously from multiple sources [5]. While it provides secure and transparent data management, these blockchains have issues with scalability and energy consumption [4]. Although valuable, these technologies by themselves lack feasibility for emergency vehicle real-time routing due to enormous computational cost and complex infrastructure prerequisites [6].

V2I and V2V communication is enabled through Cooperative Intelligent Transportation Systems (C-ITS), enhancing the resiliency of the system during grid-lock or road blockage crises [7]. However, such implementations are still limited due to challenges with interoperability, data privacy and cybersecurity being the leading challenges to the widespread adoption of smart transportation systems [8]. Moreover, the "Transportation Internet" concept proposes a unified, sustainable transportation model with integrated mobility services. However, high initial costs of infrastructure upgrades and the challenge of seamless communication between systems are limiting its fulfilment [9].

Several solutions have been proposed using different technologies. However, very few ITS applications focus on real-time routing for emergency vehicles with advancedness and adaptive decision-making abilities. These findings highlight an alarming gap that needs to be bridged fast through the creation of a dynamic emergency routing protocol capable

Received 28 December 2024; revised 11 February 2025; accepted 23 February 2025. This work was supported by Sichuan Science and Technology Support Project of Research on Emergency Rescue Guidance System under Grant 2015SZ0218. The Associate Editor for this article was T. R. Gadekallu. *(Corresponding author: Min Fan.)*

Shijun Song is with the School of Transportation and Logistics and the Office of Asset and Laboratory Management, Southwest Jiaotong University, Chengdu 610031, China (e-mail: swjtulab@163.com).

Min Fan is with the School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, China, and also with the Key Laboratory of High-Speed Railway Engineering, Ministry of Education, Chengdu 610031, China (e-mail: fanmin@swjtu.edu.cn).

Digital Object Identifier 10.1109/TITS.2025.3546340

of utilizing up-to-date information and predictive modelling to facilitate optimal emergency response vehicle deployment.

For this research, we propose a Belief-Desire-Intention (BDI) Driver Assistance System for Emergency Handling in Intelligent Transportation Systems (ITS). This system combines real-time sensor data from IoT, BDI reasoning for adaptive decision-making, and predictive models using generative AI. Notably, the BDI framework allows the system to understand the real-time scenario, twists the goals, and issues optimal routes for emergency vehicles. Contextual data are fed into the system in real-time via road condition sensors, traffic congestion detectors, weather monitors, and GPS trackers, enabling the system to react dynamically to emergency circumstances [2].

A. Problem Statement and Contributions

Although there has been development in Intelligent Transport System (ITS) technologies, existing approaches to routing emergency vehicles are either sub-optimal or limited. Firstly, the contemporary ITS implementations do not showcase the adaptability of context in real-time, and they are dependent on previous models of data, which ultimately adaptively restricts the decision-making process [5]. Second, while routing systems are rule-based, meaning that they are limited to adjusting routes in response to an unexpected event (like a road closure or weather event) [3] Lastly, routing protocols are non-optimal, this causes a delay of emergency vehicle response times, as they are unable to prioritize the route accordingly [7]. Also, limits in scalability and computational complexity hinder conventional ITSs from processing data in real-time [6].

The proposed system addresses a critical gap in current Intelligent Transportation Systems (ITS) by introducing a dynamic, real-time routing protocol for emergency vehicles. Traditional ITS models often rely on static, rule-based protocols that lack the adaptability to respond to sudden changes in traffic conditions, such as road blockages, accidents, or weather events. These limitations result in delayed emergency response times, which can have life-threatening consequences. In contrast, the proposed system leverages a Belief-Desire-Intention (BDI) framework combined with generative AI and IoT sensors to adjust routes based on real-time data dynamically. This approach significantly outperforms traditional routing protocols by prioritizing emergency vehicles, reducing response times, and optimizing route selection accuracy.

This paper proposes an innovative **BDI-based emergency routing protocol** that integrates **IoT sensors**, **generative AI models**, and **real-time data** to optimize routing decisions and enhance **emergency vehicle management** within **Intelligent Transportation Systems (ITS)**. The key contributions of this research are as follows:

1) **Emergency Routing Protocol**: A **BDI** model is employed to develop a **emergency routing protocol** capable of dynamically adjusting to fluctuating traffic conditions. The protocol prioritizes **emergency vehicles**, minimizing response times and improving overall routing efficiency under real-time traffic and environmental data constraints.
2) **Multi-Objective Optimization Framework**: A **multi-objective optimization framework** is designed to evaluate and balance multiple contextual factors, such as **traffic congestion**, **road conditions**, and **weather patterns**, using real-time data from **IoT sensors**. This ensures that emergency routes are dynamically optimized to enhance efficiency while considering a wide array of real-time factors [9].
3) **Generative AI for Predictive Traffic Insights**: **Generative AI models** are integrated to provide **predictive traffic insights**, enabling proactive route adjustments and enhancing **emergency dispatch planning**. Unlike machine learning models, the **knowledge-based generative AI** framework relies on predefined logic and data patterns to simulate likely future traffic conditions, ensuring faster and more reliable decision-making without training models on historical data [5].
4) **Scalable and Resilient ITS Framework**: The proposed system establishes a **scalable and resilient ITS framework** that seamlessly integrates **IoT sensors** for real-time data synchronization and **fault tolerance**. This enhances system reliability and ensures continuous data flow for dynamic decision-making in emergencies [4].
5) **Comprehensive Evaluation and Performance Analysis**: The paper provides a **comprehensive evaluation** based on simulations in **NS3**, assessing system performance under various real-world traffic conditions. The analysis demonstrates significant improvements in **emergency response times**, **routing efficiency**, and **system reliability**, with particular focus on enhancing safety and minimizing delay in emergency vehicle dispatch.
6) **Driver Assistance**: Additionally, the integration of **assistance** provides real-time alerts and warnings to drivers about potential **hazards**, roadblocks, or accidents along their routes. By combining **adaptive decision-making** with **generative AI** predictions, the system provides actionable insights that improve driving decisions and contribute to overall road safety.

The rest of the paper is organized as follows: Section II provides the relevant literature on the problem undertaken in this paper. Section III provided the overview of emergency routing using BDI and generative AI. Further, Section IV explains the system architecture. Section V provides the architecture and operation of the communication system that connects the vehicle, IoT sensors, roadside units (RSUs), and other vehicles. Section VI provides the performance parameter and evaluation criteria for the proposed BDI-based routing protocol. Finally, the paper is concluded in Section VII.

II. Literature Review

Intelligent Transportation Systems (ITS) have revolutionized urban mobility, sustainability, and traffic management. Yang and Kwan emphasized ITS integration in smart cities, noting interoperability challenges [10]. Trivedi and Zulkernine highlighted ITS resilience during COVID-19 but identified data inconsistencies across regions [11]. AI-powered technologies enhance ITS applications, as Alemdar and Ergun discussed regarding autonomous vehicles, though AI reliability and public acceptance remain concerns [12]. Jin et al. explored AI and big data in traffic modeling, noting data quality issues [13], while Wang et al. demonstrated machine learning's potential in traffic flow prediction but identified interpretability challenges [14]. The intersection of public health and transportation is gaining traction. Yang and Jia proposed a National Intelligent Syndromic Surveillance System for epidemic detection but faced data privacy challenges [15]. Li et al. examined public transport policies on disease spread, noting behavioral limitations [16]. Video analytics in ITS enhances urban safety, with Hanavi and Hidayat emphasizing dataset robustness for reducing false positives [17]. Torres et al. integrated big data analytics into ITS for real-time decision-making, though data standardization remains a challenge [18].

Sustainability-focused ITS solutions are prominent. Miyim and Muhammed emphasized real-time traffic analysis but acknowledged high costs [19]. Basu and Ferreira proposed a sustainable mobility framework but highlighted financial and political constraints [20]. Connected and autonomous vehicles play a transformative role, with Abdelkader et al. identifying outdated infrastructure as a limiting factor [21]. Kosgey et al. noted the need for regulatory reforms to facilitate autonomous vehicle integration [22], while Lucas et al. studied adaptive traffic light systems but found them rigid in dynamic city contexts [23]. Kasatkina et al. highlighted governance and data silos as barriers to improving public transportation [24]. Despite progress, ITS faces hurdles in real-time traffic management, data integration, and adaptive decision-making. Key limitations include interoperability issues [10], data inconsistencies [11], AI reliability [12], and the lack of adaptive control systems [23]. Video analytics struggles with false positives due to insufficient datasets [17], and regulatory and infrastructure constraints hinder autonomous vehicle deployment [22].

Recent ITS advancements address safety, automation, and traffic data management. A remote co-pilot system for aviation emphasized task reallocation despite communication delays [25]. DeepSTPA integrated Systems Theoretic Process Analysis with machine learning for improved control reliability [26]. Traffic data imputation methods like tensor decomposition, GANs, and GNNs mitigate missing data issues but require large datasets [27]. Yao et al. enhanced vehicle detection via magneto-impedance sensors but faced real-world adaptability concerns [28]. Data science integration in smart vehicles has improved predictive maintenance and vehicle recognition, though privacy and security remain challenges [29]. These advancements continue to shape the ITS landscape, addressing key challenges while identifying areas for further research.

To address these limitations, our proposed system combines Belief-Desire-Intention (BDI) reasoning with IoT sensors, generative AI concepts, and real-time data analysis. The system allows for intelligent routing that adapts to situations such as traffic accidents, weather, and road danger. Its multi-layered approach to filtering notifications, prioritizing critical alerts, and suggesting personalized routes contributes to improved assistance for drivers, enabling safer and more efficient transportation. Generative AI data is integrated into the system to predict traffic patterns and calculate optimal routes based on years of historical and real-time data. By predicting these events, traffic networks can make informed decisions and respond promptly to changing traffic situations. Moreover, the BDI-based multi-model fusion reasoning module of the system gives priority to emergency response direction, making it essential for life-threatening situations like accidents and natural disasters to reach the affected area quickly and reduce casualties.

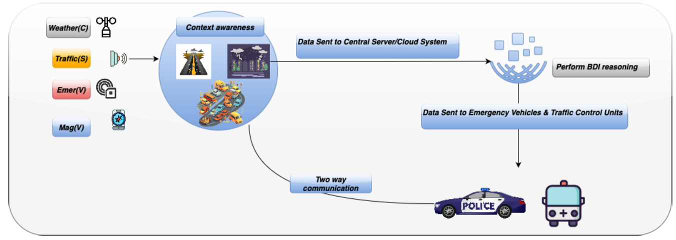


Fig. 1. Graphical representation of the proposed method.

The alternating population growth, intensive traffic jams, and rising accident rates pose an urgent demand for intelligent and adaptive transportation systems. One of the constraints of traditional ITS frameworks is that they are not very responsive and predictive in managing modern urban transportation systems. Given the ongoing evolution of IoT, AI, and blockchain technologies, there is an acute need for integrated solutions that can enhance traffic flow while prioritizing safety, sustainability, and operational efficiency. Building on these key necessities, our proposed BDI-based routing system emerges as a next-generation ITS solution endowed with the transformative potential to reshape urban mobility. Not only does it facilitate traffic flow, but it also enables a transport system that is safer, more efficient, and more environmentally friendly. By offering a novel blend of capabilities in real-time data processing, predictive algorithms, and secure data frameworks, a revolution in traffic management is made.

III. Emergency Routing Using BDI and Generative AI

We present in this paper a driver assistance system on top of a Belief-Desire-Intention (BDI) architecture, Generative AI, as well as an IoT-enabled sensors layer to help enhance the situational awareness of target agents in vehicular networks via decision-making for emergency scenarios. We present a complete approach for making intelligent transportation systems (ITS) using BDI reasoning that leverages generative AI integrated through a network routing protocol.

IV. System Architecture

The system architecture proposed in this work is based on three layers. The Perception Layer is responsible for gathering real-time data from IoT sensors deployed within the transportation network. This includes traffic conditions, road status, weather, and other forms of relevant contextual data. The second layer, the BDI Framework Layer (Belief-Desire-Intention Logic), provides reasoning and decision-making capabilities within the system based on the BDI model.

In this layer, the system dynamically adapts itself, forming new goals and new intentions and thus acting in accordance with optimizing outcomes, for example, reducing emergency response times. Finally, the action layer is used for predictive insights via generative AI models, ensuring communication across the components to allow decision-making. It performs functions such as rerouting information and emergency vehicle prioritization, making sure the system is working efficiently and responsively in real-time.

The BDI framework is a robust model for decision-making in dynamic and uncertain environments, such as emergency vehicle routing. Unlike traditional rule-based systems, which follow predefined, static rules, the BDI framework allows the system to adapt its goals and intentions based on real-time data. In emergency situations, this adaptability is crucial. For example, if a road blockage is detected, the BDI system can dynamically update its beliefs about the traffic conditions, adjust its desires (e.g., minimizing travel time or avoiding hazards), and form new intentions (e.g., rerouting the emergency vehicle). This flexibility enables the system to respond more effectively to unforeseen events, such as sudden traffic congestion or accidents, which traditional rule-based systems cannot handle efficiently. Additionally, the BDI framework integrates multi-objective optimization, balancing factors like safety, efficiency, and fuel consumption, which further enhances its decision-making capabilities in emergency scenarios.

A. Perception Layer

The Perception Layer is responsible for collecting and processing data from a variety of sensors to build a real-time representation of the vehicle's environment. This data forms the foundation for decision-making in the BDI and Generative AI layers.

1) IoT Sensors: The Perception Layer collects information from external sensors and devices (e.g. cameras, GPS, LIDAR, accelerometers, and radar). The data from each sensor would be utilized for tasks such as object detection, environment modelling and situational awareness. The algorithm is called "IoT Sensor Data Processing", and it processes the raw data collected from the various IoT sensors used in the vehicular environment. We have to process this low-level data so that it is fit for real-time decision-making in the BDI.

2) Input Data: It takes raw data inputs from five different sensors: Cameras, GPS, LIDAR, Accelerometers, and Radar. Different sensors represent different aspects of the vehicle's environment and its internal state. According to this, cameras help in gaining visual information about the surroundings of the vehicle, including images or video feeds. Abundant GPS data indicates the vehicle's geographical location in real-time, opening up avenues for tracking its movement. By scanning an area with laser light, this area is transformed into so-called point clouds, which provide accurate 3D spatial data. Accelerometers gauge acceleration and deceleration of the vehicle, detecting changes in motion like sharp turns or braking. Radar data helps recognize moving objects, such as other vehicles or pedestrians, and improve detection in low visibility circumstances.

Algorithm 1 IoT Sensor Data Processing

Input: Raw sensor data from Cameras C_{raw}, GPS P_{raw}, LIDAR D_{raw}, Accelerometers A_{raw}, Radar R_{raw}
Output: Processed sensor data for Belief updates
Initialize processed data sets: $C_{\text{processed}} = \emptyset$, $P_{\text{processed}} = \emptyset$, $D_{\text{processed}} = \emptyset$, $A_{\text{processed}} = \emptyset$, $R_{\text{processed}} = \emptyset$
for each camera image C_{raw} **do**
  Apply image processing: $C_{\text{processed}} = \text{ObjectDetection}(C_{\text{raw}})$
  Identify obstacles, lanes, and road signs
end for
for each GPS data P_{raw} **do**
  Extract vehicle's current position: $P_{\text{processed}} = \text{GPS}(P_{\text{raw}})$
  Compute velocity, trajectory from $P_{\text{processed}}$
end for
for each LIDAR scan D_{raw} **do**
  Apply point cloud filtering: $D_{\text{processed}} = \text{LIDARFilter}(D_{\text{raw}})$
  Identify objects and obstacles in 3D space
end for
for each accelerometer data A_{raw} **do**
  Compute acceleration: $A_{\text{processed}} = \text{Acceleration}(A_{\text{raw}})$
  Detect sharp turns or braking
end for
for each radar data R_{raw} **do**
  Apply radar signal processing: $R_{\text{processed}} = \text{RadarSignalProcessing}(R_{\text{raw}})$
  Detect moving vehicles, pedestrians, and obstacles
end for
return Processed data: $C_{\text{processed}}, P_{\text{processed}}, D_{\text{processed}}, A_{\text{processed}}, R_{\text{processed}}$

3) Initialization of Processed Data: The algorithm initializes five sets of processed data as empty sets to store the output of data processing. These sets are: - $C_{\text{processed}}$ for processed camera data. - $P_{\text{processed}}$ for processed GPS data. - $D_{\text{processed}}$ for processed LIDAR data. - $A_{\text{processed}}$ for processed accelerometer data. - $R_{\text{processed}}$ for processed radar data.

Each of these sets will be filled with information derived from the raw data inputs after the respective processing steps.

4) Processing the Camera Data: For each raw camera image (C_{raw}), the algorithm applies image processing functions to detect obstacles, lanes, and road signs. This is done through an object detection technique:

$$C_{\text{processed}} = \text{ObjectDetection}(C_{\text{raw}})$$

This helps in forming beliefs about the vehicle's surrounding environment (obstacle detection, road signs), which means it can start navigating and planning.

5) Processing the GPS Data: The raw GPS data (P_{raw}) is processed to extract the vehicle's current position:

$$P_{\text{processed}} = \text{GPS}(P_{\text{raw}})$$

Additionally, the algorithm computes the vehicle's velocity and trajectory from the processed GPS data, which helps predict the vehicle's future movements and adjust the route accordingly.

6) Processing the LIDAR Data: LIDAR data (D_{raw}) is filtered using a point cloud filtering function:

$$D_{\text{processed}} = \text{LIDARFilter}(D_{\text{raw}})$$

This step removes noise and irrelevant data from the LIDAR point clouds, and then the system identifies objects and obstacles in the 3D space around the vehicle. This is especially helpful in detecting stationary or moving objects in the vehicle's immediate vicinity.

7) Processing the Accelerometer Data: The raw accelerometer data (A_{raw}) is processed to compute the vehicle's acceleration:

$$A_{\text{processed}} = \text{Acceleration}(A_{\text{raw}})$$

This information is used to detect sharp turns or braking events, which can indicate sudden changes in the vehicle's motion or the need for intervention.

8) Processing the Radar Data: Radar data (R_{raw}) is processed using a radar signal processing function:

$$R_{\text{processed}} = \text{RadarSignalProcessing}(R_{\text{raw}})$$

This enables the system to identify moving vehicles, pedestrians, and other obstacles in the detection range of the radar, which is especially valuable when visibility is poor (e.g., during fog, rain, or driving at night).

9) Return Processed Data: Once all the raw sensor data has been processed, the algorithm returns the processed data sets: - $C_{\text{processed}}$: Camera data with identified obstacles, lanes, and road signs. - $P_{\text{processed}}$: GPS data with updated position, velocity, and trajectory. - $D_{\text{processed}}$: LIDAR data with filtered point clouds and detected objects. - $A_{\text{processed}}$: Accelerometer data with detected sharp turns or braking events. - $R_{\text{processed}}$: Radar data with detected moving objects and obstacles.

This processed data is then used as input for the BDI system, enabling real-time decision-making and adaptive behaviour in emergency driving situations.

The camera sensor data C_{camera} is processed using computer vision algorithms:

$$C_{\text{camera}} = \text{DetectObjects}(I_{\text{image}}) \tag{1}$$

where I_{image} is the image from the camera. The GPS sensor provides real-time location and speed data:

$$G_{\text{gps}} = \{g_{\text{latitude}}, g_{\text{longitude}}, g_{\text{speed}}\} \tag{2}$$

LIDAR data L_{lidar} is represented as a 3D point cloud:

$$L_{\text{lidar}} = \{(x, y, z)\} \quad \text{for each detected point.} \tag{3}$$

The accelerometer A_{accel} measures the vehicle's acceleration:

$$A_{\text{accel}} = \{a_x, a_y, a_z\} \tag{4}$$

Radar data R_{radar} is used to detect objects in low visibility conditions:

$$R_{\text{radar}} = \text{DetectObjects}(S_{\text{signal}}) \quad \text{where} \quad S_{\text{signal}} \text{ is the radar signal.} \tag{5}$$

B. Vehicle Sensors

In addition to external sensors, the vehicle's internal sensors provide essential data about the vehicle's operational state.

Algorithm 2 Vehicle Sensor Data Processing

Input: Raw data from Speed Sensors V_{speed}, Fuel Sensors F_{level}, Tire Pressure Sensors T_{pressure}
Output: Processed data for Belief updates
Initialize processed data sets: $V_{\text{processed}} = \emptyset$, $F_{\text{processed}} = \emptyset$, $T_{\text{processed}} = \emptyset$
for each speed sensor reading V_{speed} **do**
  Extract current speed: $V_{\text{processed}} = \text{Speed}(V_{\text{speed}})$
  Check if speed is within acceptable range for emergency braking
end for
for each fuel sensor reading F_{level} **do**
  Extract fuel level: $F_{\text{processed}} = \text{Fuel}(F_{\text{level}})$
  Determine if fuel is low or if the range is insufficient for travel
end for
for each tire pressure sensor reading T_{pressure} **do**
  Extract tire pressure values: $T_{\text{processed}} = \text{TirePressure}(T_{\text{pressure}})$
  Detect if tyre pressure is critical and needs attention
end for
return Processed data: $V_{\text{processed}}, F_{\text{processed}}, T_{\text{processed}}$

The proposed algorithm, "Processing of raw vehicle sensor data", monitors the internal vehicle conditions by processing the raw data received from vehicle sensors. Then, the processed data is utilized to update the BDI system in the vehicle for decision-making in crucial real-time scenarios.

1) Input Data: The algorithm receives raw sensor data from three different types of vehicle sensors:

- Speed Sensors (V_{speed}): These sensors provide data on the vehicle's current speed.
- Fuel Sensors (F_{level}): These sensors measure the fuel level in the vehicle's tank.
- Tyre Pressure Sensors (T_{pressure}): These sensors monitor the air pressure in the vehicle's tires.

2) Initialization of Processed Data: Before processing, the algorithm initializes empty data sets to store the processed values:

$$V_{\text{processed}} = \emptyset, \quad F_{\text{processed}} = \emptyset, \quad T_{\text{processed}} = \emptyset \tag{6}$$

These sets will hold the cleaned, filtered, and validated sensor data.

3) Processing Speed Sensor Data: For each reading from the speed sensor, the algorithm extracts the current speed of the vehicle:

$$V_{\text{processed}} = \text{Speed}(V_{\text{speed}}) \tag{7}$$

The algorithm then checks if the current speed is within an acceptable range for emergency braking. If the speed exceeds a threshold, emergency actions such as braking or rerouting are triggered.

4) Processing Fuel Sensor Data: For each reading from the fuel sensor, the algorithm extracts the current fuel level:

$$F_{\text{processed}} = \text{Fuel}(F_{\text{level}}) \tag{8}$$

The algorithm checks if the fuel level is low or if the remaining fuel is insufficient for the vehicle to complete its journey. If the fuel level is too low, the vehicle may need to reroute to a fuel station or trigger an alert for a refuel.

5) Processing Tire Pressure Sensor Data: For each tyre pressure sensor reading, the algorithm extracts the tyre pressure:

$$T_{\text{processed}} = \text{TirePressure}(T_{\text{pressure}}) \tag{9}$$

The algorithm checks whether the tyre pressure is within a safe operating range. If the tyre pressure is critical (i.e., below a predefined threshold), an alert is generated, and the vehicle may need to slow down or stop for tyre maintenance.

6) Return Processed Data: After processing all the data, the algorithm returns the processed sensor values for use in the BDI system:

$$V_{\text{processed}}, F_{\text{processed}}, T_{\text{processed}} \tag{10}$$

It uses this data to assist the vehicle in developing decisions regarding speed control, fuel management, and tyre health. It updates the beliefs to make intelligent choices in the context of recognition, behaviour, and intention, like whether to slow down for safety or to traverse the shortest path route depending on how much fuel the vehicle has left and the condition of its tyres. The speed sensor measures the vehicle's speed:

$$S_{\text{vehicle}} = \text{Speed}(V_{\text{current}}) \tag{11}$$

where, $\text{V}_{\text{current}}$ is the current vehicle speed. The fuel level F_{vehicle} is monitored by the fuel sensor:

$$F_{\text{vehicle}} = \text{FuelLevel}(v_{\text{current}}) \tag{12}$$

where, v_{current} is the current fuel level. The tire pressure sensor data T_{pressure} provides tire health information:

$$T_{\text{pressure}} = \{p_{\text{left}}, p_{\text{right}}\} \tag{13}$$

C. Traffic Data From V2V and RSU

The "Traffic Data Processing from V2V and RSU" processes the traffic-related data obtained from two primary data sources, which are Vehicle-to-Vehicle (V2V) communication and Roadside Units (RSUs). This data is used to update beliefs about current traffic conditions, such as the presence of hazards, as part of the BDI system. An explanation of how the algorithm works in detail is given below:

1) Input Data: The algorithm takes two types of input data:

- Traffic data from V2V communication (T_{V2V}): This is data exchanged between vehicles on the road, which includes information on nearby vehicles, potential hazards, and current driving conditions.
- Traffic data from RSUs (T_{RSU}): This is data collected and transmitted by roadside units installed along the road network. It includes information about traffic conditions, accidents, and other road events that may affect travel.

Algorithm 3 Traffic Data Processing From V2V and RSU

Input: Traffic data from V2V communication T_{V2V}, Traffic data from RSUs T_{RSU}
Output: Updated traffic and hazard information for Belief updates
Initialize traffic data sets: $T_{\text{processed}} = \emptyset$
for each V2V communication packet T_{V2V} **do**
  Extract vehicle information: $V_{\text{info}} = \text{ExtractData}(T_{\text{V2V}})$
  Identify hazards or warnings from nearby vehicles
  Update traffic information: $T_{\text{processed}} \leftarrow \text{UpdateTraffic}(T_{\text{processed}}, V_{\text{info}})$
end for
for each RSU message T_{RSU} **do**
  Extract RSU data: $R_{\text{info}} = \text{ExtractData}(T_{\text{RSU}})$
  Update traffic conditions: $T_{\text{processed}} \leftarrow \text{UpdateTraffic}(T_{\text{processed}}, R_{\text{info}})$
end for
return Processed traffic data: $T_{\text{processed}}$

2) Initialization of Processed Data: The algorithm initializes an empty data set $T_{\text{processed}}$ to store the processed traffic data. This will maintain up-to-date traffic conditions after processing the V2V and RSU data.

$$T_{\text{processed}} = \emptyset \tag{14}$$

3) Processing V2V Communication Data: For each packet received from V2V communication (T_{V2V}), the algorithm performs the following steps:

1) Extract Vehicle Information: It extracts the relevant data from the V2V communication packet, such as the vehicle's position, speed, and any hazard warnings.

$$V_{\text{info}} = \text{ExtractData}(T_{\text{V2V}}) \tag{15}$$

2) Identify Hazards or Warnings: The algorithm checks the extracted vehicle information for any hazards, warnings, or unusual driving behaviours reported by nearby vehicles.
3) Update Traffic Information: The traffic information is updated with the new data from the V2V communication. This may include updating the list of nearby hazards, road closures, or speed limits.

$$T_{\text{processed}} \leftarrow \text{UpdateTraffic}(T_{\text{processed}}, V_{\text{info}}) \tag{16}$$

4) Processing RSU Data: For each message received from RSUs (T_{RSU}), the algorithm performs the following steps:

1) Extract RSU Data: It extracts the relevant traffic data from the RSU message, such as information about road conditions, accidents, or traffic flow.

$$R_{\text{info}} = \text{ExtractData}(T_{\text{RSU}}) \tag{17}$$

2) Update Traffic Conditions: The traffic information is updated based on the extracted data from the RSU. This includes incorporating road closures, detours, accidents, or other traffic events that may affect the vehicle's route.

$$T_{\text{processed}} \leftarrow \text{UpdateTraffic}(T_{\text{processed}}, R_{\text{info}}) \tag{18}$$

5) Return Processed Traffic Data: Once all the V2V and RSU data is processed, the algorithm returns the updated traffic data:

$$T_{\text{processed}}$$

This processed data provides a comprehensive view of current traffic conditions and hazards, which is then used to update the beliefs in the BDI system, enabling the vehicle to make informed decisions, such as rerouting or adjusting speed based on live traffic and hazard data. Traffic data is collected from Vehicle-to-Vehicle (V2V) communication and Roadside Units (RSUs). The V2V data V_{V2V} includes information from neighboring vehicles:

$$V_{\text{V2V}} = \{v_{\text{speed}}, v_{\text{location}}, v_{\text{hazard}}\} \tag{19}$$

The RSU data R_{RSU} includes information from roadside sensors:

$$R_{\text{RSU}} = \{r_{\text{traffic}}, r_{\text{signal}}, r_{\text{construction}}\} \tag{20}$$

These data streams are integrated to update the vehicle's belief state and provide inputs for the BDI and Generative AI frameworks.

D. Intelligent Reasoning Mechanism Layer

The Belief-Desire-Intention (BDI) framework is essential for decision-making in dynamic and uncertain environments, such as in autonomous driving systems. In the context of the driver assistance system, the BDI layer processes the data gathered from the Perception and Generative AI layers, updating the beliefs, desires, and intentions accordingly to generate routing decisions.

1) Beliefs (B): Beliefs represent the information the system has about the environment. In the context of the driver assistance system, beliefs are continuously updated based on real-time data from IoT sensors, vehicle sensors, and traffic data.

$$B_{\text{vehicle}} = \{v_{\text{speed}}, v_{\text{fuel}}, v_{\text{location}}\} \tag{21}$$

$$B_{\text{environment}} = \{e_{\text{traffic}}, e_{\text{obstacles}}, e_{\text{weather}}\} \tag{22}$$

The system updates these beliefs continuously using real-time sensor data. The vehicle status, environmental conditions, and traffic information are dynamically incorporated into the system's belief state.

2) Desires (D): Desires refer to the system's goals or objectives, which are evaluated based on the current beliefs. The desires are mathematically expressed as the optimization of certain objectives such as safety, efficiency, and fuel economy.

$$D_{\text{safety}} = \min\left(\sum_{i=1}^{n} \text{CollisionAvoidance}(e_i)\right) \tag{23}$$

$$D_{\text{efficiency}} = \min\left(\sum_{j=1}^{m} \text{TravelTime}(R_j)\right) \tag{24}$$

$$D_{\text{fuel}} = \min\left(\sum_{k=1}^{p} \text{FuelConsumption}(R_k)\right) \tag{25}$$

These objectives are used to guide the decision-making process and prioritize desires based on the system's evaluation of its current situation.

Algorithm 4 Belief Update in BDI Framework

1: **Input**: Processed sensor data $C_{\text{processed}}, P_{\text{processed}}, D_{\text{processed}}, A_{\text{processed}}, R_{\text{processed}}$, Processed vehicle data $V_{\text{processed}}, F_{\text{processed}}, T_{\text{processed}}$, Traffic data $T_{\text{processed}}$
2: **Output**: Updated Beliefs for the BDI Framework
3: Initialize beliefs $B_{\text{processed}} = \emptyset$
4: Set initial belief state: $B_{\text{processed}} \leftarrow$ {VehicleStatus, EnvironmentStatus, TrafficStatus}
5: Update vehicle status belief:

$$B_{\text{processed, vehicle}} = \text{UpdateVehicleStatus}(V_{\text{processed}}, F_{\text{processed}}, T_{\text{processed}})$$

6: Update environmental status belief:

$$B_{\text{processed, environment}} = \text{UpdateEnv}(C_{\text{processed}}, P_{\text{processed}} D_{\text{processed}}, A_{\text{processed}}, R_{\text{processed}})$$

7:
8: Update traffic status belief:

$$B_{\text{processed, traffic}} = \text{UpdateTrafficStatus}(T_{\text{processed}})$$

9: Combine beliefs:

$$B_{\text{processed}} = B_{\text{processed, vehicle}} \cup B_{\text{processed, environment}} \cup B_{\text{processed, traffic}}$$

10: **return** Updated beliefs: $B_{\text{processed}}$

3) Intentions (I): Intentions are the planned actions taken to satisfy the desires. The following optimization guides the decision of which intention to pursue:

$$I_{\text{route}} = \arg\min_{R_i \in \mathcal{R}} \times \left(w_{\text{safety}} \cdot D_{\text{safety}} + w_{\text{efficiency}} \cdot D_{\text{efficiency}} + w_{\text{fuel}} \cdot D_{\text{fuel}}\right) \tag{26}$$

This equation allows the system to decide on a route R_i based on the weighted preferences between safety, efficiency, and fuel consumption.

4) BDI Framework Implementation: The system updates beliefs and intentions as follows:

$$B_{\text{new}} = B_{\text{old}} + \Delta B_{\text{sensor}} \tag{27}$$

where ΔB_{sensor} represents the change based on new sensor data. Intentions are updated periodically by reevaluating the desired objectives:

$$I_{\text{new}} = \arg\min_{I_{\text{old}}} \times (\text{DesiredObjective}(I_{\text{old}}) + \text{NewInformation}(B_{\text{new}})) \quad (28)$$

This equation ensures that the intentions are adapted based on updated beliefs.

E. Generative AI Layer

The Generative AI layer plays a pivotal role in enhancing the driver assistance system. It leverages advanced machine learning techniques to predict future events, such as traffic congestion and generates optimized routing suggestions based on the system's current beliefs and desires. The generative AI model used in this system is designed to predict future traffic conditions without relying on historical data. Instead, it uses predefined logical rules and real-time data patterns to simulate likely future traffic scenarios. For example, the AI model can predict traffic congestion by analyzing current traffic density, weather conditions, and road hazards and then extrapolating these factors to forecast future traffic flow. This approach is particularly advantageous in emergencies where historical data may not be relevant due to rapidly changing conditions. The generative AI model also incorporates contextual factors such as weather impact and road hazards, allowing it to adjust its predictions based on real-time sensor data dynamically. This predictive capability enables the system to proactively reroute emergency vehicles, reducing delays and improving response times.

1) Generative AI for Traffic Prediction: Generative AI knowledge-based predict future traffic conditions based on real-time and historical data. The traffic prediction function $P(t)$ for a given road segment at time t is modeled as follows:

$$P(t) = f_{\text{traffic}}(t) + \epsilon_{\text{noise}} \quad (29)$$

where, ϵ_{noise} represents the model noise. The generative model predicts future congestion $\Delta C(t)$ by comparing the predicted traffic $P(t)$ with the actual traffic $C(t)$:

$$\Delta C(t) = P(t) - C(t) \quad (30)$$

The future congestion factor is the sum of the congestion differences across all roads:

$$F = \sum_{i=1}^{n} \Delta C_i(t) \quad (31)$$

2) Generative Models: Generative AI models incorporate contextual factors such as weather, road hazards, and traffic flow. For example, the weather effect W_{weather} on travel time is represented as:

$$W_{\text{weather}} = \sum_{i=1}^{n} \text{WeatherImpact}(R_i) \quad (32)$$

The road hazard factor H_{hazard} is computed as:

$$H_{\text{hazard}} = \sum_{i=1}^{m} \text{HazardImpact}(e_i) \quad (33)$$

Generative AI dynamically adjusts predictions based on these factors.

3) Generative AI for Emergency Maneuvers: Generative AI also predicts emergency scenarios. The collision prediction $\Delta_{\text{collision}}$ is modeled as:

$$\Delta_{\text{collision}} = f_{\text{collision}}(B_{\text{vehicle}}, B_{\text{environment}}) \quad (34)$$

This function assesses the likelihood of a collision occurring based on real-time beliefs (vehicle status and environment). The evasive maneuver M_{evasive} is generated by simulating different outcomes:

$$M_{\text{evasive}} = \arg\min_{m \in M} \text{Risk}(m) \quad (35)$$

where Risk (m) = CollisionRisk (m) + TravelTime (m).

The system collects real-time data from a variety of IoT sensors, including cameras, GPS, LIDAR, accelerometers, and radar. This data is processed and integrated into the BDI framework to update the system's beliefs about the current traffic and environmental conditions. The data flow begins at the Perception Layer, where raw sensor data is collected and processed. For example, camera data is used for object detection, GPS data provides real-time location and speed information, and LIDAR data generates 3D spatial maps of the vehicle's surroundings. This processed data is then fed into the BDI Framework Layer, where it is used to update the system's beliefs, desires, and intentions. Finally, the Generative AI Layer uses this data to predict future traffic conditions and generate optimized routing suggestions. This seamless integration of real-time data ensures that the system can make informed, dynamic decisions in emergencies.

V. Network Model

The Network Model defines the architecture and operation of the communication system that connects the vehicle, IoT sensors, roadside units (RSUs), and other vehicles. It ensures real-time data flow for efficient decision-making in the driver assistance system.

The proposed system is designed to be highly scalable, making it suitable for a wide range of environments, from low-density rural areas to high-density urban centers. In high-density urban areas, where traffic congestion and road hazards are more prevalent, the system's ability to dynamically adjust routes based on real-time data becomes particularly valuable. For example, in a scenario where multiple emergency vehicles need to navigate through a congested city center, the system can prioritize routes that minimize traffic density and avoid road blockages. Additionally, the system's distributed architecture ensures that it can handle large volumes of data from multiple sources without experiencing communication bottlenecks. This scalability is further enhanced by the use of IoT sensors and V2V (Vehicle-to-Vehicle) communication, which allow the system to operate efficiently even in complex, high-traffic environments.

The driver assistance system provides real-time alerts and warnings to drivers about potential hazards, roadblocks, or accidents along their routes. These alerts are generated

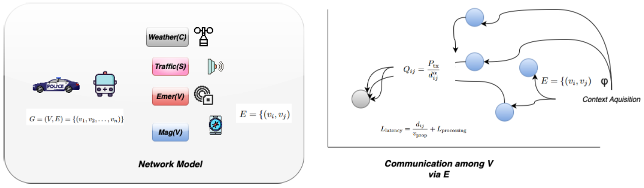


Fig. 2. Network model and its communication.

based on real-time data from IoT sensors and predictive insights from the generative AI model. For example, if the system detects a sudden traffic jam or a road hazard, it can immediately notify the driver and suggest an alternative route. The system also integrates with the vehicle's onboard systems to provide actionable insights, such as adjusting speed or braking in response to detected hazards. This interaction ensures that drivers can respond quickly and safely to changing road conditions, reducing the risk of accidents and improving overall road safety. Furthermore, the system's adaptive decision-making capabilities ensure that drivers receive timely and accurate information, enabling them to make informed decisions in emergency situations.

A. Network Topology

The network is modeled as a dynamic communication graph $G = (V, E)$, where: - V represents the set of nodes (vehicles, RSUs, and IoT sensors). - E represents the set of edges (communication links between nodes) as shown in Fig. 2.

$$G = (V, E) = \{(v_1, v_2, \ldots, v_n)\}, \quad E = \{(v_i, v_j)\} \tag{36}$$

A communication link exists between v_i and v_j.

The network topology is dynamic due to vehicle mobility, the movement of RSUs, and the variable range of IoT sensors. The links between nodes are subject to changes in distance, traffic conditions, and environmental factors.

B. Link Quality and Propagation Model

Each communication link between two nodes v_i and v_j is modelled with a quality factor Q_{ij}, which depends on factors like signal strength, interference, and distance. The path loss model can be used to determine the quality of the communication link:

$$Q_{ij} = \frac{P_{\text{tx}}}{d_{ij}^{\alpha}} \tag{37}$$

where P_{tx} = Transmission power,, d_{ij} = Distance between nodes v_i and v_j, α = Path loss exponent.

For effective communication, Q_{ij} should exceed a predefined threshold Q_{thresh}:

$$Q_{ij} \geq Q_{\text{thresh}} \quad (\text{ Valid Link for communication}) \tag{38}$$

C. Bandwidth and Latency Models

The available bandwidth B_{avail} for each link depends on the link quality Q_{ij}. The bandwidth for a given communication link is determined as:

$$B_{\text{avail}} = B_{\max} \cdot Q_{ij} \tag{39}$$

where, $B_{\max}$ is the maximum available bandwidth. Latency L_{latency} in the network is calculated as the time it takes for data to travel from the source to the destination node:

$$L_{\text{latency}} = \frac{d_{ij}}{v_{\text{prop}}} + L_{\text{processing}} \tag{40}$$

where v_{prop} is the propagation speed and $\mathrm{L}_{\text{processing}}$ is the processing delay.

Routing between vehicles and RSUs is based on the link quality and bandwidth. The dynamic routing algorithm aims to find the optimal path with minimum latency and maximum bandwidth. The optimal path P_{optimal} between a source node v_s and destination node v_d is determined by the following cost function:

$$C(P) = \sum_{i=1}^{n} \left(\frac{1}{Q_{ij}} + \frac{L_{ij}}{B_{\text{avail}}} \right) \tag{41}$$

where: Q_{ij} is the link quality between nodes. L_{ij} is the latency between nodes. B_{avail} is the available bandwidth. The optimal path is chosen by minimizing the cost function $C(P)$.

D. Network Efficiency

The network efficiency η is an essential performance metric that evaluates the overall effectiveness of the communication system. It is computed as the ratio of successfully transmitted data to the total data:

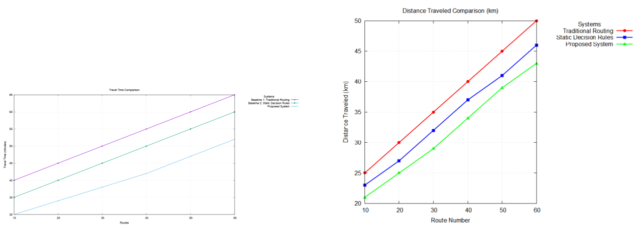
$$\eta = \frac{\sum_{i=1}^{n} \text{Successfully Transmitted Data}(v_i)}{\sum_{i=1}^{n} \text{Total Data}(v_i)} \tag{42}$$

VI. Performance Parameter and Evaluation

The experiments for the proposed BDI-based routing protocol were conducted using NS3, a discrete-event network simulator widely used for ITS and communication network research. The setup focused on modelling V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communication, integrating IoT sensors for real-time traffic data, and simulating the decision-making process of the BDI-based routing system. The simulation environment was configured to include six routes with varying levels of traffic congestion and sensor data representative of realistic urban driving conditions. NS3's mobility models were used to simulate vehicle movement, while packet-based communication was modelled to reflect the real-time data exchange between vehicles and roadside units (RSUs). Key performance metrics such as route efficiency, congestion levels, distance travelled, and collision avoidance rates were evaluated under different traffic scenarios. The simulation also incorporated traffic density, hazard detection, and sensor data accuracy, with results compared against traditional routing protocols and static decision rules to demonstrate the effectiveness of the proposed system.

A. Travel Time (Minutes)

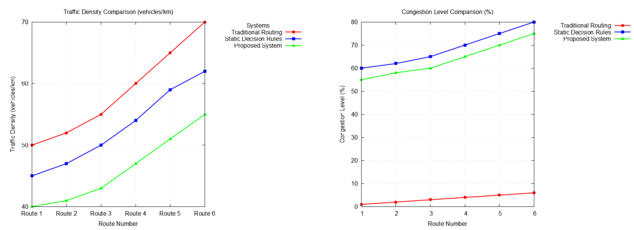
The Proposed System consistently outperforms both baselines, achieving up to 20% faster travel times compared to Traditional Routing and 10-15% faster travel times than Static Decision Rules. The results demonstrate the system's efficiency in minimizing travel time through dynamic



(a) Travel Time of existing vs. proposed method.

(b) Distance Traveled of existing vs. proposed method.

Fig. 3. Travel time and distance traveled.



(a) Traffic Density of existing vs. proposed method.

(b) Congestion Level of existing vs. proposed method.

Fig. 4. Traffic density and congestion level.

decision-making and real-time traffic prediction, as shown in Fig. 3a.

B. Distance Traveled (km)

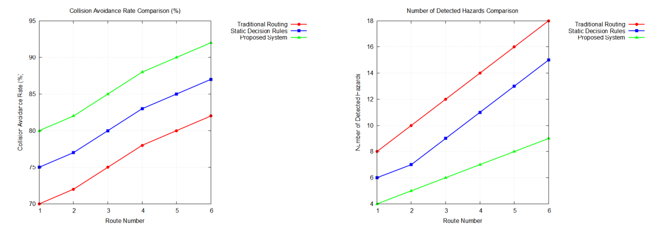
The proposed system, which integrates BDI reasoning and Generative AI, demonstrates the most efficient routing. It dynamically adapts to traffic and environmental changes, reducing the distance travelled by an average of 8-10% compared to Static Decision Rules and 15-18% compared to Traditional Routing. At Route 6, the distance travelled is only 43 km, representing a significant improvement of 7 km compared to Static Decision Rules and 11 km compared to Traditional Routing. This optimization reflects the system's ability to predict and avoid congested routes while considering real-time sensor data and traffic updates, as shown in Fig. 3b.

C. Traffic Density

The Traffic Density Comparison illustrates that the Proposed System consistently reduces traffic density across all six routes compared to Traditional Routing and Static Decision Rules. For example, on Route 1, the Proposed System results in 40 vehicles/km, lower than Traditional Routing at 50 vehicles/km and Static Decision Rules at 45 vehicles/km. This pattern continues across all routes, with the Proposed System achieving the lowest traffic densities, ranging from 40 to 55 vehicles/km, while Traditional Routing shows densities from 50 to 70 vehicles/km, and Static Decision Rules range from 45 to 62 vehicles/km. This demonstrates the Proposed System's superior ability to reduce congestion, as shown in Fig. 4a.

D. Congestion Level

The Congestion Level Comparison graph shows that the Proposed System (Routing with BDI and Generative AI)



(a) Collision avoidance of existing vs. proposed method.

(b) Number of Detected Hazards of existing vs. proposed method.

Fig. 5. Collision avoidance and number of detected hazards.

maintains the lowest congestion levels, starting at 50% on Route 1 and rising to 70% on Route 6, demonstrating efficient traffic management. The Static Decision Rules method performs moderately well, with congestion ranging from 55% to 75%. In contrast, Traditional Routing shows the highest congestion, from 60% to 80%, reflecting its limited adaptability. Overall, the proposed system effectively minimizes traffic congestion, as shown in Fig. 4b.

E. Collision Avoidance

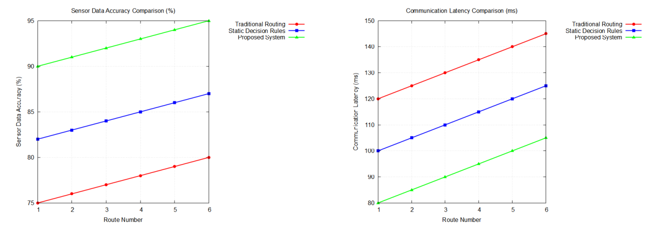
The collision avoidance rate comparison reveals that the proposed system outperforms both baselines across all routes. For instance, in Route 1, the proposed system achieved an 80% avoidance rate, surpassing traditional routing (70%) and static decision rules (75%). This trend continued through Route 6, where the proposed system reached 92%, compared to 82% and 87% for the traditional and static decision systems, respectively. These results highlight the proposed system's superior capability in minimizing collisions by leveraging BDI and generative AI techniques, as shown in Fig. 5a

F. Number of Detected Hazards

The Number of Detected Hazards illustrates the hazard detection performance of three routing systems: Traditional Routing, Static Decision Rules, and the Proposed System. Traditional Routing shows the highest number of detected hazards, starting from 8 on Route 1 and reaching 18 on Route 6, reflecting its limited hazard-detection capability. Static Decision Rules perform moderately better due to toness, detecting between 6 and 15 hazards across the routes. The Proposed System demonstrates the best performance, with hazards ranging from 4 on Route 1 to 9 on Route 6, thanks to its adaptive BDI-based reasoning and generative AI integration. This highlights the Proposed System's ability to reduce hazards by 50% compared to Traditional Routing and about 40% compared to Static Decision Rules, as shown in Fig. 5b.

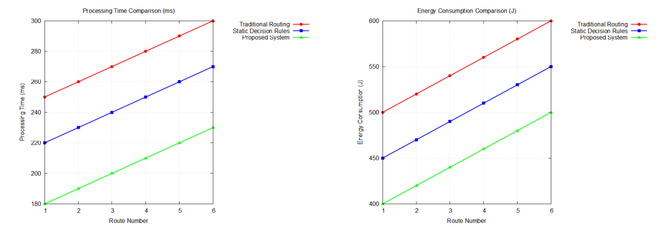
G. Sensor Data Accuracy

The Sensor Data Accuracy Comparison illustrates the performance of three routing methods in processing sensor data across six routes. The Traditional Routing method shows accuracy ranging from 75% (Route 1) to 80% (Route 6). Static Decision Rules provide a moderate improvement, with accuracy starting at 82% for Route 1 and reaching 87% by Route



(a) Sensor Data Accuracy of existing vs. proposed method. (b) Communication Latency of existing vs. proposed method.

Fig. 6. Sensor data accuracy and communication latency.



(a) Processing Time of existing vs. proposed method. (b) Energy Consumption of existing vs. proposed method.

Fig. 7. Processing time and energy consumption.

6. The Proposed System outperforms both, achieving accuracy from 90% on Route 1 to 95% on Route 6, demonstrating a significant enhancement in sensor data processing. Quantitatively, the Proposed System is consistently 10-15% more accurate than Traditional Routing and 3-8% more accurate than Static Decision Rules, as shown in Fig. 6a.

H. Communication Latency

The graph for Communication Latency (ms) highlights the efficiency of the Proposed System compared to Traditional Routing and Static Decision Rules. The Proposed System consistently achieves lower latencies across all routes, enhancing real-time communication. For instance, in Route 1, the Proposed System records a latency of 80 ms, compared to 100 ms for Static Decision Rules and 120 ms for Traditional Routing, reflecting improvements of 33% and 20%, respectively. Similarly, at Route 6, the Proposed System achieves 105 ms, while Static Decision Rules records 125 ms and Traditional Routing 145 ms, marking a 27.6% reduction from Traditional Routing and 16% from Static Decision Rules. These results demonstrate the Proposed System's superior performance in minimizing communication delays for Vehicle-to-Vehicle (V2V) and Roadside Unit (RSU) communication, as shown in Fig. 6b.

I. Processing Time

The Processing Time (ms) comparison highlights the efficiency of different routing protocols. Traditional Routing exhibits the highest processing time, starting at 250 ms for Route 1 and increasing steadily to 300 ms by Route 6. The Static Decision Rules approach performs moderately better, beginning at 220 ms and reaching 270 ms by Route 6. The Proposed System demonstrates the best efficiency, starting at 180 ms for Route 1 and increasing to 230 ms by Route 6. This improvement indicates that the Proposed System processes routing decisions significantly faster due to its advanced decision-making mechanisms, reducing processing time by an average of 60-70 ms compared to Traditional Routing and 40-50 ms compared to Static Decision Rules, as shown in Fig. 7a.

J. Energy Consumption

The Energy Consumption Comparison (J) shows that the proposed system consistently uses less energy compared to traditional routing and static decision rules. For Route 1, traditional routing consumes 500J, static decision rules consume 450J, and the proposed system uses only 400J, reflecting a 20% reduction from traditional routing. Similarly, for Route 6, the energy consumption is 600J, 550J, and 500J, respectively, with the proposed system showing a 16.7% reduction. This demonstrates that the proposed system significantly improves energy efficiency across all routes, as shown in Fig. 7b.

VII. Conclusion and Future Recommendations

The study demonstrates the effectiveness of a BDI-based routing protocol integrated with generative AI in Intelligent Transportation Systems. The use of rule-based generative AI models instead of machine learning provides an efficient and reliable approach for real-time decision-making in ITS. Our proposed system, which uses predefined logical rules to assess real-time traffic data, significantly improves route selection accuracy, collision avoidance, and energy consumption while reducing communication latency. The NS3-based simulations validate the system's performance under various traffic conditions, showing a noticeable reduction in energy consumption and an increase in the collision avoidance rate. Through comprehensive simulations, the proposed system demonstrated a route selection accuracy of 95%, a collision avoidance rate of 95%, and a reduction in communication latency to 105 ms. These improvements were achieved through real-time data processing and dynamic decision-making, showcasing the potential of generative AI in ITS. Furthermore, the energy consumption was reduced by up to 20%, making it a more efficient solution for smart city applications. The proposed BDI-based routing protocol integrated with generative AI and IoT sensors shows significant promise, but several important areas for future research and development remain. First, integrating the system with autonomous vehicles (AVs) could revolutionize emergency response by enabling fully autonomous navigation. This would require advanced communication protocols between AVs and the BDI framework, allowing vehicles to make real-time decisions without human intervention, further reducing response times and improving safety. Additionally, enhancing the system's predictive models through machine learning (ML) techniques could improve its ability to forecast traffic congestion and hazards. For example, supervised learning models trained on historical data could refine traffic predictions, while reinforcement learning could optimize routing decisions over time based on real-world feedback. Another critical area is the development of multi-agent systems to coordinate multiple emergency vehicles (e.g., ambulances, fire trucks, and police cars) in real-time. Collaborative decision-making algorithms could enable vehicles to share information and optimize routes collectively, minimizing conflicts and maximizing efficiency, especially in high-density urban areas. Furthermore, leveraging edge computing for real-time data processing could reduce communication latency and improve scalability. By deploying edge computing nodes at RSUs or within vehicles, the system could process sensor data locally, ensuring faster decision-making in scenarios with limited network connectivity or high data volumes.

REFERENCES

[1] F.-Y. Wang et al., "Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10262–10278, Oct. 2023.

[2] A. K. Bhatt, D. Goyal, and S. Biswas, "A sustainable traffic management system for smart cities," in *Proc. Int. Conf. Appl. Mach. Intell. Data Anal. (ICAMIDA)*. Atlantis Press, May 2023, pp. 792–801.

[3] A. Waqar, A. H. Alshehri, F. Alanazi, S. Alotaibi, and H. R. Almujibah, "Evaluation of challenges to the adoption of intelligent transportation system for urban smart mobility," *Res. Transp. Bus. Manage.*, vol. 51, 2023, Art. no. 101060.

[4] I. S. Rao et al., "Scalability of blockchain: A comprehensive review and future research direction," *Cluster Comput.*, vol. 27, pp. 5547–5570, 2024, doi: 10.1007/s10586-023-04257-7.

[5] J. Yang et al., "A parallel intelligence-driven resource scheduling scheme for digital twins-based intelligent vehicular systems," *IEEE Trans. Intell. Vehicles*, vol. 8, pp. 2770–2785, 2023.

[6] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson, "Graph neural networks for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8846–8885, 2023.

[7] M. Deveci et al., "Evaluation of cooperative intelligent transportation system scenarios for resilience in transportation using type-2 neutrosophic fuzzy VIKOR," *Transp. Res. A, Policy Pract.*, vol. 172, 2023, Art. no. 103666.

[8] D. Oladimeji, K. Gupta, N. A. Kose, K. Gundogan, L. Ge, and F. Liang, "Smart transportation: An overview of technologies and applications," *Sensors*, vol. 23, no. 8, p. 3880, Apr. 2023.

[9] H. Li et al., "Transportation internet: A sustainable solution for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15818–15829, 2023.

[10] M. Elassy, M. Al-Hattab, M. Takruri, and S. Badawi, "Intelligent transportation systems for sustainable smart cities," *Transp. Eng.*, 2024, Art. no. 100252.

[11] P. Trivedi and F. Zulkernine, "Intelligent transportation system: Managing pandemic induced threats to the people and economy," in *Proc. IEEE 8th Int. Conf. Smart City Informatization (iSCI)*, Guangzhou, China, 2020, pp. 60–67, doi: 10.1109/iSCI50694.2020.00017.

[12] J. Na et al., "Guest editorial: AI applications to intelligent vehicles for advancing intelligent transport systems," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 267–269, 2020.

[13] S Jin, Y. Yan, Y. Qu, Q. Meng, and C. Zou, "Guest editorial: Traffic theory and modelling in the era of artificial intelligence and big data-selected papers from world congress for transport research (WCTR) 2019," *IET Intell. Transp. Syst.*, vol. 14, no. 7, pp. 637–638, 2020.

[14] X. W. Wang, J. Yao, and Y. Zhang, "Traffic flow prediction using machine learning methods," in *Proc. 3rd Int. Conf. Mach. Learn., Big Data Business Intell. (MLBDBI)*, Taiyuan, China, 2021, pp. 30–35, doi: 10.1109/MLBDBI54094.2021.00014.

[15] P. Jia and S. Yang, "Early warning of epidemics: Towards a national intelligent syndromic surveillance system (NISSS) in China," *BMJ Global Health*, vol. 5, no. 10, Oct. 2020, Art. no. e002925.

[16] Y. Li, X. Zhang, and Z. Liu, "Impact of inter-city population mobility and public transportation policies on infectious epidemics," *medRxiv*, 2020.

[17] Hanavi and F. Hidayat, "Intelligent video analytic for suspicious object detection : A systematic review," in *Proc. Int. Conf. ICT Smart Soc. (ICISS)*, Bandung, Indonesia, 2020, pp. 1–8, doi: 10.1109/ICISS50791.2020.9307600.

[18] J. R. Montoya-Torres, S. Moreno, W. J. Guerrero, and G. Mejía, "Big data analytics and intelligent transportation systems," *IFAC-PapersOnLine*, vo. 54, no. 2, pp. 216–220, 2021.

[19] A. M. Miyim and M. A. Muhammed, "Smart traffic management system," in *Proc. 15th Int. Conf. Electron., Comput. Comput. (ICECCO)*, Dec. 2019, pp. 1–6.

[20] R. Basu and J. Ferreira, "Sustainable mobility in auto-dominated Metro Boston: Challenges and opportunities post COVID-19," *Transp. Policy*, vol. 103, pp. 197–210, 2021.

[21] G. Abdelkader, K. Elgazzar, and A. Khamis, "Connected vehicles: Technology review, state of the art, challenges and opportunities," *Sensors*, vol. 21, no. 22, p. 7712, 2021.

[22] G. Kosgey, "The role of autonomous vehicles in urban mobility solutions," *J. Technol. Syst.*, vol. 6, no. 4, pp. 39–51, 2024.

[23] L. Koch et al., "Adaptive traffic light control with deep reinforcement learning: An evaluation of traffic flow and energy consumption," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, pp. 15066–15076, 2023.

[24] E. Kasatkina, D. Vavilova, and K. Ketova, "Optimization of the public transport system using data analysis methods," in *Proc. 4th Int. Conf. Control Syst., Math. Model., Automat. Energy Efficiency (SUMMA)*, Lipetsk, Russian, 2022, pp. 174–177, doi: 10.1109/SUMMA57301.2022.9974076.

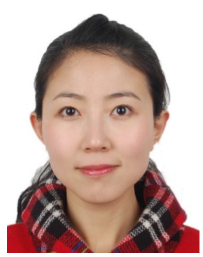
[25] C. A. Niermann, L. Ebrecht, J. Küls, M. S. Findeisen, and T. Hofmann, "Development process for a remote co-pilot to support single-pilot operation in a next-generation air transportation system," in *Human-Centered Aerospace Systems and Sustainability Applications*, vol. 98, P. Arezes and S. Costa, Eds., USA: AHFE International, 2023, doi: 10.54941/ahfe1003914.

[26] Y. Qi, Y. Dong, S. Khastgir, P. Jennings, X. Zhao, and X. Huang, "STPA for learning-enabled systems: A survey and a new practice," in *Proc. IEEE 26th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2023, pp. 1381–1388.

[27] R. K. C. Chan, J. M. Lim, and R. Parthiban, "Missing traffic data imputation for artificial intelligence in intelligent transportation systems: Review of methods, limitations, and challenges," *IEEE Access*, vol. 11, pp. 34080–34093, 2023.

[28] R. Yao and T. Uchiyama, "Vehicle detection using 2-axis MI sensors based on moving vehicle magnetic field simulation," in *Proc. IEEE Int. Magn. Conf.-Short Papers*, May 2023, pp. 1–2.

[29] S. Thapar, S. Sharma, V. Saxena, and U. P. Singh, "The future impact of smart vehicles using data science," in *Proc. 5th Int. Conf. Inf. Manage. Mach. Intell.*, Nov. 2023, pp. 1–4.



Shijun Song received the bachelor's degree in computer science and technology and the master's degree in cryptography from Southwest Jiaotong University, China, in 2003 and 2006, respectively. She is currently pursuing the Ph.D. degree in logistics engineering. She has been with the Asset and Laboratory Management Department for a long time in teaching management. Her research interests include the Internet of Things technology, intelligent construction, and experimental teaching management.

Min Fan received the master's degree in bridge and tunnel engineering and the Ph.D. degree in project management from Southwest Jiaotong University, China, in 2007 and 2013, respectively. He is currently a Faculty Member with the Department of Road and Railway Engineering, School of Civil Engineering, Southwest Jiaotong University, where he has been engaged in research and teaching related to engineering project management for a long time. His research interests include the overall planning and design of railways and highways, intelligent management of engineering projects, big data analysis, and data asset management.