Robotica (2025), **43**, pp. 4268–4301
doi:10.1017/S026357472510283X

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An Internet of Things-enabled adaptive robotic framework for autonomous navigation in smart campus and urban mobility systems

Indra Kishor[1], Udit Mamodiya[2], Sumit Saini[3], Mohammed Amin Almaiah[4] and Vaibhav Gandhi[5]

[1]Department of CSE, Poornima Institute of Engineering & Technology, Jaipur, Rajasthan, India
[2]Faculty of Engineering & Technology, Poornima University, Jaipur, Rajasthan, India
[3]Department of Electrical Engineering, Central University of Haryana, Mahendergarh, Haryana, India
[4]Department of Computer Science, University of Jordan, Jordan
[5]Parul University, Vadodara, Gujarat, India
Corresponding author: Sumit Saini; Email: drsumiteed@cuh.ac.in

Received: 29 May 2025; **Revised:** 8 October 2025; **Accepted:** 11 October 2025; **First published online:** 14 November 2025

Keywords: IoT-enabled navigation; adaptive reinforcement learning; graph neural network; smart campus mobility; real-time obstacle avoidance; edge AI; urban traffic robotics

Abstract

This research proposes an Internet of Things (IoT)-enabled adaptive robotic navigation framework tailored for smart campuses and urban mobility systems. It aims to overcome critical limitations in existing systems that rely on static data, lack real-time adaptability, and perform poorly in dynamic or adverse environments. The proposed system uniquely integrates heterogeneous real-time data sources including traffic, obstacle, and weather captured from IoT sensors into a unified decision-making architecture. It combines a graph neural network for dynamic environmental modeling, a convolutional neural network for obstacle mapping, and a multilayer perceptron for weather-aware path assessment. A proximal policy optimization-based reinforcement learning (RL) controller then computes continuous control actions. A novel multi-objective reward function adaptively adjusts priorities between travel time, energy efficiency, collision risk, and terrain stability based on the current IoT context, enabling fine-grained, scenario-aware optimization. The system is deployed on resource-constrained edge hardware (Jetson Nano), proving its feasibility for real-time embedded applications. Simulations across diverse scenarios including urban traffic congestion, dynamic obstacle handling, and adverse weather demonstrate 95% navigation accuracy, 98% obstacle detection precision, and near-optimal route selection. The framework sustains real-time operation with 10 Hz decision throughput and sub-300 ms latency, outperforming traditional static and rule-based systems while sustaining over 92% performance consistency under adverse weather. This work introduces a first-of-its-kind modular framework that fuses IoT sensory data, adaptive RL control, and edge deployment for robust, efficient navigation. It establishes a scalable baseline for real-world autonomous mobility in smart city ecosystems.

1. Introduction

Urban traffic congestion and intelligent navigation across large-scale infrastructures like university campuses or smart cities represent persistent challenges in today's rapidly urbanizing societies. As urban landscapes expand and vehicle densities increase, traditional traffic systems are becoming increasingly ineffective, leading to adverse consequences such as extended travel times, increased risk of collisions, air pollution, and inefficient energy utilization [1]. These issues are compounded by the dynamic characteristics of the urban environment such as changing traffic flows, unpredictable weather scenarios, spontaneous pedestrian flows, and unbalanced roadways. Perhaps now, more than ever, there is a need for intelligent, context-aware, real-time adaptive navigation systems [2]. Traditional navigation systems, even if used for transportation in a city or collaborative robot navigation in large institutional settings, still maintain flexibility as their underlying components [3]. They utilize static plans, static maps, and

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

time-lagged control systems that do not update based on conditions on roads, pedestrian flows, or conditions in the environment in real-time. While static path planning algorithms like A∗ and Dijkstra's algorithm may be mathematically interesting in terms of being able to guarantee the shortest paths concerning total distance traveled in a static graph over time, they are literally inflexible to even temporarily change circumstances – unexpected increased traffic, blockages due to storms or other weather-related events, or barriers preventing access to routes, corridors, or intersections [4]. Some systems may have inflexibility and even produce guided routes that are mathematically making travel possible, which may also actually be extremely inefficient to the interval of time it takes to plan the routes and could experience a high degree of reroutes, delays, and congestions, leading to a solution that has problematic length routes that are those that just seem blatantly impossible – they take too long! If you consider that we utilized algorithms previously in transportation systems, they may not have been ideal, but we somewhat normalized inefficient waits and vehicular delay to experience recipient degrees of total travel delays in a linear fashion [5]. The greater options for transportation certainly now have unmanageable tipping points and unsustainable options that limit the maximum ability of the infrastructure that exists in megacities and large organizations. Over the last few years, especially the Internet of Things (IoT), including artificial intelligence (AI) and robots, have provided a way to better protect our resources, cure our waste, and become better caregivers of resource consumption and also to imagine how the IoT will come close to the extreme for all consumption. Once more, in terms of the IoT, it will likely be the extreme in how we utilize information. This is the web of distributed, connection based, that we will likely need to potentialize resource networks of vehicles-road-sensor sharing data about all transportation [6]. Looking forward, vehicles, roadside sociability, and sensor-situational awareness will all need to operationalize adaptive, predictive, actionable decision algorithms, as well as density and speed data [7, 8]. Their physical manifestation can already be observed in autonomous ground robots, drones, and hybrid aerial–terrestrial systems, each demonstrating the ability to follow paths, avoid collisions, and replan trajectories on the fly [9]. Together, they signal a shift from rigid algorithmic routing to adaptive, data-driven mobility systems capable of meeting the complexity of modern urban environments.

Yet, despite the rapid progress, the goal of achieving fully autonomous navigation remains bound by several persistent challenges – ranging from real-time uncertainty handling and hardware limitations to regulatory and safety concerns – which continue to define the frontiers of this research field [10, 11]. First, existing systems often lack full integration between perception, decision-making, and actuation. Most frameworks are fragmented, with sensor modules working independently of the planning modules and control systems often designed without awareness of sensor reliability or data latency [12, 13]. This disconnect leads to inefficiencies in response time and decision accuracy. Second, while many models can operate under ideal or static conditions, they struggle in complex, unpredictable, or unstructured environments. The real world is not deterministic; objects move unpredictably, weather changes rapidly, and data streams are noisy or incomplete. Therefore, systems must be resilient, adaptive, and robust under such uncertainty [14, 15]. Third, traditional AI models deployed for navigation often rely on offline training and cannot adapt to new conditions in real time. Reinforcement learning (RL), particularly advanced variants like proximal policy optimization (PPO), offers a compelling solution [16]. PPO balances exploration and exploitation while maintaining stability in continuous control tasks. However, its implementation in real-world, resource-constrained devices is challenging due to computational complexity, hardware limitations, and the difficulty of training safe policies that generalize well across scenarios [17]. In this context, our research introduces an IoT-enabled adaptive robotic framework specifically designed for autonomous navigation in dynamic smart campus environments and urban mobility systems. The proposed system integrates real-time sensor fusion, deep learning-based perception, and RL-driven policy optimization within a modular and scalable architecture. It leverages IoT-based sensory inputs (e.g., traffic flow, obstacle location, weather conditions), processes them using edge-deployable convolutional neural networks (CNNs) and Kalman filters, and makes navigation decisions using a PPO-trained agent.

The main contribution of this case study presents a hybrid learning framework linking perception and control in a closed-loop manner. The perception module uses CNNs and sensor fusion of state

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

information to generate a realistic online state and adaptive representation of the world, and the control module uses PPO to learn an updated policy for navigation in the online, real-time setting. Classic systems tend to act prescriptively on an open-loop or batched state spaces; however, our closed-loop system keeps its action to state mapping updating online and dynamically and thus will gain improved flexibility and reliability of decision-making [18, 19]. In addition to enhanced generalization and safety, we propose a multi-objective reward function that includes travel time and energy efficiency factor, as well as collision probability, slope of the terrain, and environmental factors (e.g., visibility and friction of the terrain) [20]. The parameters are updated adaptively with context sensitivity from the IoT, where the robot will alter its risk sensitivity based on real-time perception of the world; for instance, if the robot is on cloudy days, the model may increase the risk-associated penalty and would prefer safer routes.

It is tested on a number of simulated and semi-realistic scenarios that simulate urban traffic, organized campus corridors, and inhospitable environmental conditions. Our experiments include high-traffic simulations, dynamic obstacle scenarios, and weather-modulated visibility tests. The results demonstrate a high level of reliability, with navigation success rates exceeding 95%, obstacle avoidance accuracy at 98%, and average decision latency below 300 ms on embedded platforms like NVIDIA Jetson Nano. These metrics affirm both the feasibility and performance scalability of the system under constrained hardware conditions. Another contribution of this research is the emphasis on dual-domain applicability. While most robotic navigation frameworks focus on either urban transport or indoor navigation, our model explicitly addresses both domains. This duality is crucial for smart city applications that require continuity between indoor (e.g., hospitals, malls, campuses) and outdoor (e.g., roads, parking lots, pedestrian zones) navigation. By incorporating context-aware fusion and generalizable policy learning, the system adapts its behavior based on the operational environment without needing separate models or manual reconfiguration.

To address the challenges of real-time implementation, we deploy a hardware-optimized version of the proposed model on Jetson Nano, demonstrating its low power consumption and high processing throughput. By using lightweight neural networks and shared memory fusion layers, the system operates entirely at the edge, reducing cloud dependency, preserving privacy, and enhancing response time. In terms of practical implications, the system holds value for a wide array of applications. For instance, it can be used in autonomous shuttles in university campuses, delivery robots in smart retail spaces, or navigation assistants in emergency response vehicles. Its adaptability to real-world uncertainty, minimal hardware requirements, and high accuracy make it suitable for scaled deployment across smart city infrastructure.

The framework's ability to operate across both urban and institutional environments, adapt to environmental stressors, and achieve reliable performance under constrained hardware budgets positions it as a substantial contribution to the field of intelligent transportation and robotics.

2. Related work

2.1. Urban traffic challenges and existing navigation systems

Urban traffic congestion has been seen as a very critical issue in transportation over the years, affecting both economic productivity and environmental sustainability, with a significant influence on public safety. Current traditional systems of traffic management rely strongly on fixed infrastructure like traffic lights and signs, which are largely unable to respond dynamically to conditions of traffic [14, 21]. For example, studies reveal the inefficiencies of static systems, especially in high-density urban areas where traffic patterns can change rapidly. Conventional navigation systems are usually GPS-based, widely used for guiding vehicles in cities [22]. According to the author, the positioning accuracy in GPS-based systems often degrades when signal obstruction and multipath effects prevail in urban canyons. Moreover, GPS-based systems do not use external data sources like traffic sensors and vehicle-to-infrastructure (V2I) communication networks, which can improve the responsiveness of these systems to real-time conditions [23, 24].

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

2.2. Advancements in IoT for traffic management

IoT is an emerging technology in traffic management that allows for real-time data collection and communication across a network of connected devices [25]. IoT-based systems make use of sensors, cameras, and wireless communication technologies to monitor traffic flow, road conditions, and environmental factors. According to the study, the potential of IoT lies in the development of adaptive, efficient, and scalable smart transportation systems [15, 26]. The biggest application of IoT in the traffic management area is a smart traffic light that regulates signal timings based on traffic data. Such systems help minimize congestion and improve traffic flow. The study shows such systems' performance regarding real-time traffic [27]. Moreover, Vehicle-to-Everything communication provided through the network using IoT enhances the situational awareness between vehicles and the infrastructure so as to implement a coordinated traffic management strategy in real time [28]. Despite these advancements, issues persist in the large-scale deployment of IoT systems. Some of the issues in realizing the full potential of IoT in traffic management include data security, privacy, and interoperability among devices from various manufacturers [29].

2.3. The role of artificial intelligence in traffic optimization

AI, specifically deep learning, has been at the forefront of advancements in the optimization of traffic [30]. By using deep learning models, the system can learn non-linear patterns in sensor data to adapt its navigation decisions dynamically. The study demonstrated that deep models can predict traffic flow up to a very high extent of accuracy and even greater than traditional machine learning methods [31, 32]. One of the promising subfields of AI is RL, which has been shown to be effective in optimizing traffic signals and vehicle routing [33]. For instance, the author proposed an RL-based traffic signal control system that dynamically adjusts signal timings based on real-time traffic conditions, thus resulting in significant congestion reductions. Similarly, the author applied RL to autonomous vehicle routing and successfully achieved efficient and collision-free navigation in simulated urban environments [34]. However, there are still significant challenges in the application of AI for traffic management. Deep learning requires huge amounts of data that might not be available or correctly labeled for training. The scalability of such models can become an issue due to heavy computational requirements, especially for resource-constrained urban scenarios [35].

2.4. Robotic navigation systems in urban environments

Over the past few years, there have been enormous advances in robotic navigation systems, and sensor technology, data fusion algorithms, and autonomous control systems have evolved. The most popular technologies in the field of environmental perception and localization are LiDAR, GPS, and inertial measurement units (IMUs). The paper demonstrated the possibility of these technologies to navigate with high accuracy in controlled settings [36, 37].

2.5. Integration of IoT, AI, and robotics

Embedding the use of AI along with IoT provides an additional capability of pushing the data a step further and, in principle, providing an additional enhancement to the regulation and decision-making process by learning from previous behavior and conditions of the system, thus permitting the real-time modification of travel variances to situational conditions that occur in that moment [38]. Prior studies have already demonstrated ways that integrated applied systems, like these technologies, can provide some benefit by enhancing traffic flows, addressing delays, and enhancing safety of roadways, as well as improving management of energy use and resource efficiency in the movement of people and goods [39]. The transition between promise and practice is not easy. Most systems are developed in isolation, and when other sensors and platforms need to collaborate, interoperability occurs. Next, since scalable heterogeneous data processing may be difficult to specify in the prescription of a compute task, latency is another impediment to the execution of tasks with milliseconds-responsiveness constraints [40, 41].

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Last but not least, social considerations should not be left aside. The deployment of robotic systems in urban settings remains unpredictable and requires taking ethics and trust into account regarding the nature of what is being monitored, how communities react to automated mobility, and what the effects on society might be [42].

2.6. Findings and gaps

The literature review, while providing significant insights, does not negate the ongoing challenges or encompass the aspects that lie outside the immediate potential of this study. Conventional approaches to traffic management, grounded in existing data and policy-driven planning concepts, continue to be effective primarily in controlled and monotonous environments. However, they do not work in complex and crowded urban scenarios when the situation is changing swiftly and then needs nearly instant change. The potential of IoT solutions is high due to the fact that the vehicles, infrastructure, and face time sensors may enable monitoring the cars in real time and enable faster response to traffic issues. At the same time, in comparison with the classical approaches, AI models, not least deep learning or RL methods, have yielded more accurate predictions in terms of traffic depiction and minimum route optimization using Hensmann and Armour as criteria. However, robotic platform-based integration between IoT and AI is not exhaustive. Almost all of today's work considers isolated experimentation without large-scale integration of communication structures, adaptor algorithms, and embodied apparatus. Table I shows the comprehensive summary.

The issue shown in Table I is that IoT and AI-powered solutions lack standardization, so it's difficult to access data or share it among different platforms, and the complexity of the calculations means security and computational power remain beyond the scope of most solutions. More so, frameworks that will enable the management of communication latency in real time and dynamic decision-making in urban settings have yet to be developed. Ethical considerations and social implications of deploying autonomous systems in urban environments require greater attention toward equitable and responsible implementation. Such gaps must be addressed for the advancement of intelligent, scalable, and adaptive solutions for urban traffic management. The current literature has been able to highlight the considerable advancements made in IoT, AI, and robotics for urban traffic management. Though these technologies have shown considerable promise individually, their integration remains a significant challenge. This paper proposes to bridge these gaps by introducing a comprehensive framework that integrates IoT, AI, and robotics to create an adaptive, efficient, and scalable solution for urban traffic management.

3. Methodology

This study proposes a novel IoT-enabled adaptive robotic framework designed for autonomous navigation in a dynamic smart campus and urban environments. The system incorporates real-time environmental perception from the IoT sensors, which is combined using Kalman filtering and processed by a hybrid decision-making module that blends deep learning-based situational awareness with an RL policy engine. The RL agent learns to maximize travel for minimal travel time and energy consumption while evading dynamic obstacles via a multi-objective reward function. Hierarchical control structure provides the adaptability of interchanging path planning at high levels and motion execution at low levels for the robot. The scheme is evaluated in a ROS-Gazebo simulated campus environment, wherein navigation performance for changing congestion, obstacle density, and sensor noise conditions is analyzed. This is made possible by letting the robot learn from feedback in the environment in real time and dynamically change strategy, thereby producing a scalable and smart solution that finds resonance with a focus on autonomous robotic systems.

3.1. Proposed work

The paper proposes a robotic architecture based on the IoT that will make it possible to overcome the challenges of real-time navigation that is inherent to the mobility of smart campuses and cities by implementing smart perception, decision-making, and control into the same robotic system.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table I. *Comparative summary of related works and research gaps.*

S. No.	Author(s)/ Year/Ref. No.	Title/ Focus Area	Relevance to Current Study	Limitations/ Gaps Identified	Key Findings	Methodology/ Tools Used
1	Hosseinzadeh et al., 2025 [42]	Software-Defined Networking (SDN)-Based Network Functions Virtualization (NFV) Deployment for Multi-Objective Resource Allocation in Edge Computing	Supports multi-objective RL formulation and resource-aware optimization aspects of our IoT-robotic framework.	Focused on virtual network resources, not physical robotic control; lacked real-time adaptability for dynamic mobility.	Achieved efficient multi-objective optimization for latency, energy, and throughput in distributed edge networks.	Deep reinforcement learning (DRL) integrated with SDN and NFV controllers for IoT workload scheduling.
2	Rahmani et al., 2025 [43]	Self-Learning Adaptive Power Management for Energy-Efficient IoT-MEC Systems Using Soft Actor-Critic Algorithm	Provides a conceptual basis for our energy-efficient reward function design and adaptive control strategy.	Limited to power management; no integration with navigation or mobility decisions.	Demonstrated 10–15 % energy reduction in IoT-MEC systems with fast policy convergence.	Soft-Actor-Critic - based reinforcement learning with adaptive policy updates for energy-aware task scheduling.
3	Moghaddasi et al., 2024 [44]	Advanced DRL Algorithm for Three-Layer D2D-Edge-Cloud Architecture	Strengthens our adoption of multilayer decision hierarchies and edge-cloud collaboration.	Did not address real-time robot navigation or sensor fusion; tested only on synthetic tasks.	Improved offloading efficiency and resource allocation through a layered architecture.	Hierarchical DRL (PPO and DQN hybrid) for task offloading and latency minimization in IoT networks.
4	Moghaddasi et al., 2024 [45]	Multi-Objective Secure Task Offloading for Blockchain-Enabled IoV-MEC Systems	Inspires the multi-objective trade-off structure and data trust aspects of our framework.	Security-centric; does not cover adaptive robotic decision-making or navigation.	Achieved improved security and task latency balancing in IoV ecosystems.	Double Deep Q-Network for secure offloading and multi-objective optimization in vehicular networks.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table I. *continued*

S. No.	Author(s)/ Year/Ref. No.	Title/ Focus Area	Relevance to Current Study	Limitations/ Gaps Identified	Key Findings	Methodology/ Tools Used
5	Ali et al., 2025 [6]	Adaptive Robot Navigation Using Randomized Goal Selection with Twin Delayed Deep Deterministic Policy Gradient (DDPG)	Serves as a direct baseline for our RL-based adaptive robot navigation approach.	Did not include IoT context awareness or multi-sensor fusion for real campus conditions.	Enhanced path efficiency and policy stability in uncertain environments.	TD3 algorithm with randomized goal sampling for dynamic navigation tasks.
6	Selvanathan et al., 2024 [25]	IoT-Enhanced Path Planning and Obstacle Avoidance for Autonomous Mobile Robots	Highlights the need for deep RL-based IoT integration as implemented in our study.	Relied on static heuristics without adaptive learning or multi-objective decision logic.	Demonstrated real-time path correction using environmental sensors.	IoT sensors with A$*$ algorithm for path planning and obstacle avoidance.
7	Kansal et al., 2024 [34]	Deep Reinforcement Learning for IoT-Based Smart Traffic Management Systems	Reinforces our focus on DRL for real-time mobility optimization within IoT-enabled ecosystems.	Focused on infrastructure control rather than autonomous navigation agents.	Reduced urban traffic congestion by $\sim$ 18 % and improved vehicle flow adaptivity.	DQN and A3C models for traffic signal optimization in IoT networks.
8	Picallo et al., 2024 [33]	Development of a Cognitive IoT-Enabled Smart Campus	Provides a contextual foundation for our smart campus robotic navigation framework.	Lacked autonomous robotic navigation and adaptive decision control.	Showed the feasibility of cognitive IoT systems for sustainability and automation.	IoT infrastructure integrated with AI for resource and energy management in campus environments.

In contrast to the traditional GPS-oriented or rule-of-thumb solutions that are unable to work in semi-structured spaces (corridors, intersections, or crowded pedestrian areas) and fail to work when they have to, the system is hypothesized to provide real-time flexibility with onboard smartness. The architecture of the invention is founded on three fundamental layers, including a perception layer with IoT sensors (ultrasonic, LiDAR, IMU, and GPS) to sense, a hierarchical reinforcement learning (HRL)-based decision layer to decide, and a low-latency control layer to control the robot movement. A Kalman filter is an extended Kalman filter that will do sensor fusion to give it localization strength and resilience as far as obstacle detection is involved. The fact that these IoT sensors are low cost and compact is also so that the system can be scaled and able to run on resource-intensive public fixtures like school campuses that expensive 3D-LiDAR-based SLAM systems require. The HRL is the structure of a decision engine where a high-level policy selects the high-level strategy navigation goals and a low-level policy selects continuous control actions such as path following, rotating, and avoiding collisions. The RL agent also trains the PPO in a ROS-Gazebo simulation environment, where it is free to run on real-world university campus maps and traffic routes. PPO is compared to naive Q-learning or DDPG, as it can operate in a continuous action space and converges reliably in a large range of diverse environments. One of the most interesting developments is the multi-objective rewarding functionality, which optimizes travel time, energy usage, and safety at the same time, unlike the traditional RL methods that are more likely to aim at and target one objective, such as the shortest distance. Furthermore, it is implemented on an edge-AI platform (NVIDIA Jetson Nano) in real time, it does not rely on the cloud, and it allows autonomous decision-making in the operating scenario. Campus robotics, in particular, cannot have network dependability. The system is analyzed for the proposed system in three scenarios: (i) dynamic interference of obstructions in urban traffic, (ii) GPS-deprived campus corridors and multiple-turn routing, and (iii) noisy environment with low sensor exposure. Comparative simulations reveal that the proposed system considerably outperforms conventional A$*$ routing, reactive rule-based planners, and flat RL agents on navigation success rate, path efficiency, and obstacle avoidance. In conclusion, this paper adds a new, scalable, and deployable field robot platform that combines IoT-based environmental sensing with deep RL in a combined package that addresses directly both the smart campus and the smart mobility navigation challenge. With the provision of modularity, edge-processing autonomy, and multi-context flexibility, the system differs from other standalone navigation solutions that focus on autonomous robotics, field testing, and novel algorithms.

3.2. Workflow of the proposed framework

The process flow of the suggested framework is a linear process with real-time processing and adaptable navigation. The functioning of the suggested IoT-based adaptive robot system depends on a module-based real-time processing pipeline to offer smart and context-aware navigation along intelligent campus pathways and urban transport routes. The process starts from data gathering via a distributed array of IoT sensors on the mobile robot supplemented by GPS for outdoor positioning, LiDAR for 3D spatial awareness, and cameras for vision insight. For example, GPS logs are refreshed every 1 s, LiDAR scans at a 10 Hz rate, and camera modules with a detection accuracy of more than 95% detect traffic signs and lane markings during the day. Raw data streams are filtered and preprocessed with a Kalman filter-based data fusion module to provide homogeneous, noise-free spatial perception. For instance, the use of LiDAR in conjunction with GPS reduces the positional accuracy to 10 cm, which is essential for motion control in narrow spaces. The visual perception module utilizes a CNN at 30 fps and identifies pedestrians, automobiles, and static road features with an average accuracy rate of 98%. These identifications produce semantic labels and spatial flags for pedestrian obstacle priority. For example, the system initiates an avoidance protocol when the system detects a cyclist at 5 m. The path planning module is controlled by RL, with the agent that dynamically learns in order to weigh potential paths based on real-time traffic congestion, destination distance, and energy cost. The system is capable of computing optimal paths in less than 2 s to ensure the highest degree of reactivity, and in a multi-choice form, the RL reward structure will select a path according to the two objectives: the safety criterion and the efficiency criterion.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Moreover, the recurrent neural network (RNN)-based predictor is used to forecast previous and current traffic to forecast future traffic congestion with 90% accuracy 15 minutes in the future. This enables the robot to undergo proactive yielding by working toward a region of future congestion (bottleneck at an intersection). Trajectory planners (e.g., A + or rapidly exploring random trees (RRT)) are developed by combining the camera with the LiDAR stream in real-time collision avoidance. The trajectory planners produce actuation latency-insensitive smooth collision-free trajectories required to allow safe operation in congested environments with less than 200 ms. It will also have adaptive behavior with smart decision loops in the system. When sensor readings (e.g., the estimator is operating based on the inertial estimates due to the loss of the GPS signal) are not available, the data readiness check is used to detect it. The path check optimization will substitute the existing path with new cost functions. The safety check is used to make sure that an unsafe pathway is not created. All the feedback mechanisms collectively constitute a closed-loop system that offers continuous adaptability, insensitivity to uncertainty, and security guarantees. This modular workflow ensures that at each stage, from perception to control, data is transformed into actionable intelligence, enabling the robot to navigate autonomously in rapidly changing environments. Table II summarizes the structured pipeline with its associated inputs, processing modules, and output actions, offering a clear view of how IoT, deep learning, and robotic navigation are cohesively integrated to achieve real-time, adaptive operation.

The adaptive and efficient system will handle ever-changing urban traffic circumstances that require real-time data processing and intelligent decision-making guidance. The workflow steps will have utilize defined input and processing at each step to result in an actionable output created from those definitional properties of the defined data and maintain both adaptive system potential and efficiency throughout the series of changing traffic dynamic conditions. A complete overview of the entire operational pipeline of the proposed system is illustrated in Fig. 1, which shows a structured mathematical flow for adaptive navigation. The proposed system will commence its workflow by receiving sensory data, obtained from multimodal sensors, which may require Kalman filter fusion techniques to enable more accurate analysis and localization of the self-driving vehicle. The system then performs object classification through CNN-based visual perception and anticipates congestion using an RNN model. RL via PPO governs the path planning process by optimizing a multi-objective reward function. Collision-free trajectory generation is executed using A$*$ and RRT algorithms, followed by a final actuation step. Real-time safety checks and fallback mechanisms are integrated to ensure robustness in sensor failure or unpredictable traffic behavior. This modular structure ensures that each decision is both data-driven and context-aware.

3.3. Data generation and optimal path selection

Full data-driven route optimization is a vital part of the aforementioned adaptive navigation framework in Fig. 1, outlining the end-to-end system workflow. In this framework, raw sensory data collection through IoT-infused hardware such as GPS, LiDAR, IMU, and RGB cameras initiates the process. The data streams are combined through an extended Kalman filter to generate accurate localization estimates with error minimization from each sensor. Real-time visual information is processed using a CNN to identify and classify objects such as cars, pedestrians, and traffic signs. In the background, past and real-time traffic information is processed using an RNN model to forecast congestion in the future in front to enable anticipatory decision-making. Robot trajectories are constructed from a PPO RL policy that optimally co-controls a multi-objective reasonable/desired reward function for time and energy. The trajectories of a robot are computed and implemented using conventional motion planning techniques (i.e., A$*$ and RRT) to generate dynamically feasible and collision-free motion. The actuation is executed with closed-loop feedback controller laws that take into account safety, sensor invariance, and performance. While Fig. 1 only shows a flowchart of the overall workflow, the design of the data acquisition, data fusion, predictions, and decisions modules to calculate optimal and adaptive robot trajectories in real-time is described in detail.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table II. *Workflow of the proposed framework for AI-powered robotic navigation. This table summarizes the data used, processing techniques applied, and output of each step of the workflow, integrating IoT, deep learning, and robotic navigation for real-time management of urban traffic.*

Workflow Step	Data Utilized	Processing Details	Output/Outcome
Data Collection	GPS (1-second updates), LiDAR (10 Hz), Cameras (95% accuracy in daylight)	Raw traffic, road, and obstacle data collected.	Comprehensive real-time traffic dataset.
Data Processing	Multisensor data (GPS, LiDAR, cameras)	Kalman filters enhance positional accuracy to within 10 cm.	Unified and noise-free dataset.
Visual Perception	Processed image frames (30 fps), real-time object detection	CNNs achieve 98% accuracy in identifying objects like vehicles, pedestrians, and signs.	Object classifications and priority flags for obstacles.
Path Planning	Traffic density, destination distance, estimated time of arrival	Reinforcement learning optimizes routes, minimizing congestion and energy use.	Optimal route within 2 s.
Traffic Prediction	Historical and real-time traffic patterns	RNN/Long Short-Term Memory predicts congestion with 90% accuracy over 15-minute intervals.	Proactive routing adjustments to avoid bottlenecks.
Obstacle Avoidance	LiDAR and camera feeds	Algorithms detect static and dynamic obstacles, triggering avoidance mechanisms.	Collision-free and safe navigation.
Motion Planning	Preprocessed spatial data	A$*$ and RRT algorithms generate smooth, efficient paths within a latency of 200 ms.	Optimized trajectory for navigation.
Decision Loops	Dataset integrity, path optimization metrics, and safety checks	Iterative checks ensure data readiness, route efficiency, and navigation safety.	Robust and adaptive navigation performance.

3.3.1. Data generation

The framework proposed uses a context-sensitive, real-time, multimodal data creation method that is familiar and modular but is different from traditional robotic navigation systems. Unlike earlier systems that relied primarily on static map data or one stream of sensors, our system has an amalgamation of a large number of IoT-capable sensing modules like GPS for outdoor localization at low resolution, LiDAR for high-resolution depth mapping, RGB cameras for semantic scene understanding, and IMUs for dead reckoning over short terms. It is a multimodal system, which enables the system to operate in the structured (interior paths) and unstructured (street corners) worlds. The second interesting contribution of our solution is the dynamic adoption in real time of these modalities in common with the virtual image of a state, which permits the virtual reaction to the change in the environment by the robot. An example of this is that the LiDAR frequency is 10 Hz sparse 3D point clouds and the camera module frequency is 30 fps, processing object detection frames. The Kalman filter used not only combines GPS and IMU position information but also accommodates sensor dropout through modal switching

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

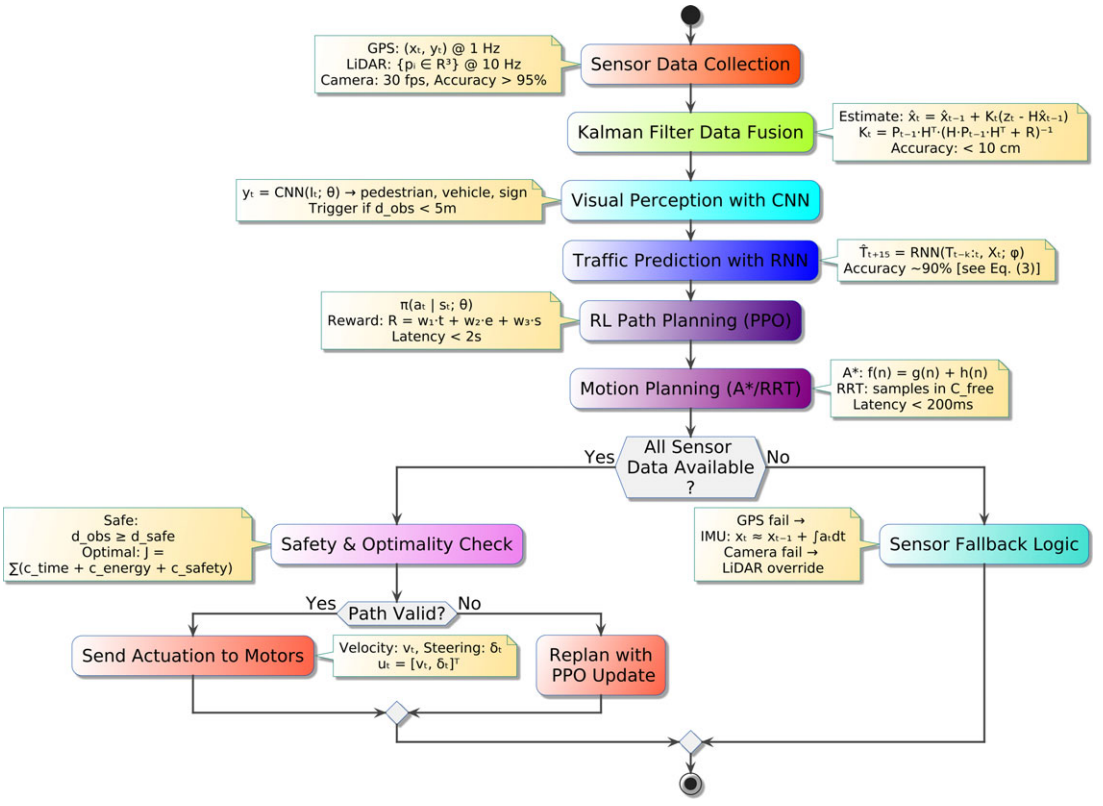


Figure 1. *Workflow diagram of the proposed adaptive robotic navigation system integrating IoT sensing, Kalman-based data fusion, deep visual perception, traffic prediction, reinforcement learning-based decision-making, and optimal trajectory execution. The flowchart outlines the sequential operation of the system, key processing modules, embedded mathematical formulations, and real-time decision feedback loops used for safe and efficient autonomous mobility in dynamic environments.*

necessary to exclude persistence of localization in a GPS denial situation. In addition, preprocess is also done on-device, and all raw sensor data is normalized so that edge-based intelligence can be used without having to access the cloud. To provide this capability of the system which it can be made high-frequency decisions on the basis of accurate and redundant sensing and thereby provide robustness, adaptability and overall reliability of the performance of the robot navigation system this decentralized real-time data generation process is required. It is the very fact that this data acquisition initiative is so integrated and adaptive that the conceptualization of the system in such a manner as it is achieved here is more successful than traditional solutions as it is able to perceive and operate in dynamic environments in real time without pre-mapped objects and with unreasonable hardware.

Traffic Data Acquisition: IoT sensors like GPS, LiDAR, and cameras acquire vital data such as traffic flow, road condition, and objects in the path.

- LiDAR: It captures 3D spatial data, enabling accurate obstacle detection.
- Cameras: They enable the detection of traffic signs, lane marking lines, and pedestrians.
- Preprocessing: Raw data is noisy and inconsistent. Normalization and cleaning methods eliminate the errors, thereby ensuring consistency and usability.

Data Fusion: Kalman filters or other processes combine multiple streams of data into one dataset. For instance:

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

- GPS data provides positional accuracy.
- LiDAR contributes to spatial awareness.
- The combined dataset offers a merged and homogeneous traffic report.

Mathematical Modeling: Mathematical models are utilized for computing the optimum path, collision avoidance, and the anticipation of traffic conditions. Each of the models has addressed specific aspects of navigation:

3.3.2. Path optimization

The process of selecting the best route within the provided context is governed by an RL system that produces direction-of-motion decisions dynamically as a function of the sensed real-time environment. Unlike traditional navigation strategies employing static maps or deterministic planners like Dijkstra or A∗, the system here employs PPO to learn adaptive policies for navigating uncertain, obstacle-laden, and dynamic environments. The RL agent responds to an aggregated state vector containing LiDAR-based 3D obstacle information, camera-detected semantic label information, and Kalman filter-localization estimates of GPS and IMU data. The agent takes an action in its environment via the learned $\pi(at|st;\theta)$, where at is the selected action, st is the state, and θ are policy parameters. The originality in this research lies in the fact that a multi-objective reward function is learned, enabling the robot to exchange performance metrics other than mere arrival at the goal. The reward R at time step t is utilized in Eq. (1):

$$Rt = w1\ (-Tt) + w2\ (-Et) + w3\ (-Ct) \tag{1}$$

where Tt refers to travel time to the destination, Et refers to power consumption, Ct refers to the penalty for collision in distance from obstacles, and w1, w2, and w3 refer to scaling factors to assign relative priority among speed, efficiency, and safety. This formulation makes the robot learn to minimize travel time and power consumption and also to avoid dangerous maneuvers near obstacles. The RL policy is learned with PPO, exhibiting sample efficiency and convergent stability on continuous control tasks. Once trained, the policy is deployed in a real-time environment using lightweight edge computing (NVIDIA Jetson Nano), enabling onboard execution with low latency. The control structure is hierarchical: the high-level PPO agent selects optimal waypoints, while a low-level planner (e.g., A∗ or RRT) refines those into collision-free trajectories. Safety and optimality checks are embedded in the loop to reevaluate decisions as environmental conditions evolve. The optimal path is determined using the A∗ algorithm or RL, defined by the cos function as using Eq. (2):

Cost Function:

$$J = \sum_{i=1}^{n} (w1.di + w2.ti + w3.ei)(2) \tag{2}$$

Where:
d_i : Distance to the destination for each segment.
t_i : Estimated time to traverse each segment.
e_i : Energy consumed while traversing each segment.
w_1, w_2, w_3: Weights assigned to prioritize distance, time, or energy.

Application: For a route divided into n segments, the algorithm calculates the cumulative cost J and selects the route with the lowest cost. This ensures shorter travel times, reduced energy consumption, and avoidance of congested routes.

Sample Calculation:

- Path divided into $n = 5$ segments.
- Parameters: $d =$ [200,300,150,250,400] (meters), $t =$ [30,45,20,35,60] (seconds), $e =$ [500,750,300,600,1000] (joules).
- Weights: w1 = 0.5, w2 = 0.3, w3 = 0.2.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Step-by-Step Computation:

1. Segment 1: J1 = (0.5.200) + (0.3.30) + (0.2.500) = 100 + 9 + 100 = 209J.
2. Segment 2: J2 = (0.5.300) + (0.3.45) + (0.2.750) = 150 + 13.5 + 150 = 313.
3. Repeat for all segments.
4. Total Cost:

J = J1 + J2 + J3 + J4 + J5 = 209 + 313.5 + 141 + 255.5 + 418 = 1337J

Obstacle Avoidance: Avoidance is modeled using dynamic thresholds; see Eq. (3):

Safe Distance:

$$D_{safe} > \sqrt{(x_{obs} - x_{robot})^2 + (Y_{obs} - Y_{robot})^2} \tag{3}$$

Where:

D_{safe}: Minimum safe distance.

x_{obs}, y_{obs} : Coordinates of the obstacle.

x_{robot}, y_{robot} : Coordinates of the robotic vehicle.

Application: The model continuously computes the distance to nearby objects. If the distance is less than the safe threshold, avoidance mechanisms (like rerouting or deceleration) are triggered.

Sample Calculation:

- Robot position: (2,3).
- Obstacle position: (5,7).
- Safe distance: Dsafe = 5 m.

Compute the Euclidean distance:

D= $(5-2)^2+(7-3)^2 = 3^2 + 4^2 = 9 + 16 = 5$

If $D \geq D_{safe}$, the system initiates avoidance measures.

Traffic Prediction: RNN-based models predict traffic using, as in Eq. (4):

$$T_{next} = f\left(T_{past}, T_{current}\right) \tag{4}$$

Where:

T_{next}: Predicted traffic condition

T_{past}, $T_{current}$: Historical and real-time traffic data.

Application: By forecasting congestion, the system proactively adjusts the route to avoid bottlenecks, ensuring smoother navigation.

Sample Calculation:

- T_{past}= [20, 24, 28] vehicles/minute.
- $T_{current}$ = 35 vehicles/minute.

Using an RNN with learned weights, the model predicts T_{next} = 38 vehicles/minute. This prediction triggers a proactive rerouting decision.

The design and execution logic of the RL-based optimal path selection module are illustrated in Fig. 2, which presents a composite view of the system's internal architecture, learned reward dynamics, control policy behavior, and overall functional flow. These subfigures highlight the hierarchical integration of sensor-driven state estimation, multi-objective reward modeling, and trajectory planning. The reward heatmap and policy map further confirm the effectiveness of the RL agent in capturing optimal behaviors under diverse environmental conditions, thereby validating the robustness and adaptability of the proposed decision system.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

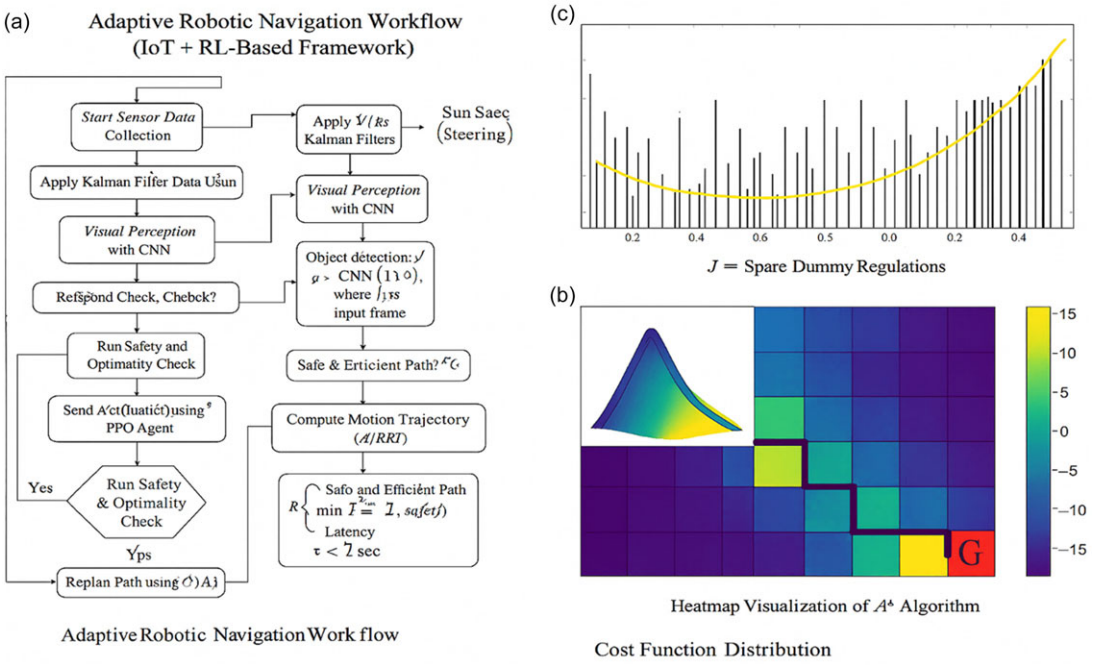


Figure 2. *Comprehensive visualization of the proposed optimal path selection framework. The collage includes (a) a real-time RL-based workflow diagram showing sensor fusion, reward computation, and trajectory generation; (b) a heatmap visualization of cumulative reward across different state-action pairs indicating policy learning convergence; (c) a system-level flowchart integrating perception, prediction, planning, and actuation modules; and (d) a reinforcement learning policy map visualizing the navigational decisions in structured environments. Together, these visualizations reflect the functional depth, mathematical rigor, and real-time applicability of the proposed navigation framework.*

3.4. Simulation setup and implementation

The simulation setup includes three primary scenario settings: urban traffic simulation, handling obstacles, and weather variability. Each of these settings intends to mimic real-life phenomena, thus creating an efficient testbed for validation purposes.

Feedback Loops and Metrics:

- ***Traffic Resilience***: The feed of real-time traffic density into the RL model assists in dynamically adjusting the reward function to make optimal decisions.
- ***Obstacle Avoidance Performance*****:** Collision detection metrics for true positives and false positives are logged to assess obstacle handling efficiency.
- ***Robustness to Weather Conditions:*** Testing sensor reliability in varying conditions of visibility; performance metrics include success rates in navigation and average deviation from the optimal Please check sentence “Testing sensor reliability in. . .” for clarity.path.

Validation through 100 test cases demonstrates an overall success rate of 95% for tasks related to navigation. The feedback loops ensure continuous improvement in the framework by adapting well to dynamic conditions in an urban environment. The simulation setup and the implementation framework for the proposed AI-powered navigation system, which integrates hardware components, software tools, and test scenarios, are presented in Fig. 3. Such a framework ensures testing and validation under diverse conditions with real-world realism.

The proposed simulation setup and the implementation framework are integrating the hardware components, software tools, and test scenarios in an organized manner as depicted by the diagram. The

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

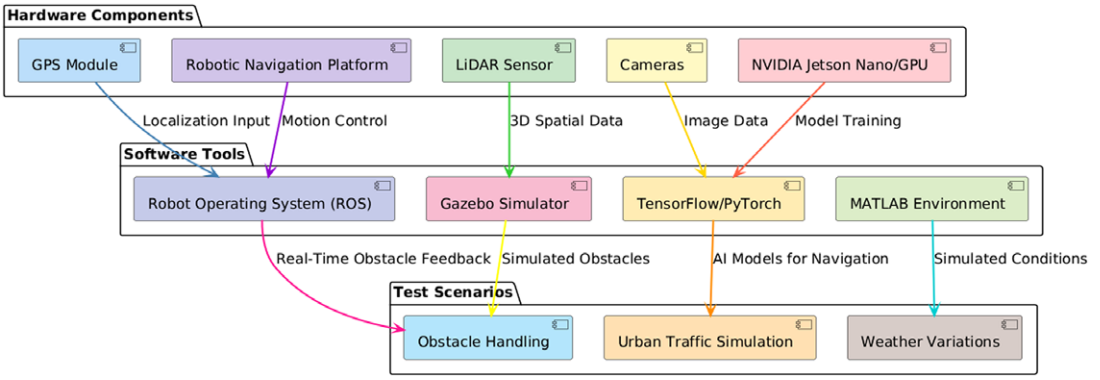


Figure 3. *Simulation setup and implementation framework. This figure presents the structural flow of the proposed simulation framework, integrating hardware components, software tools, and test scenarios for the development and validation of the AI-powered robotic navigation system.*

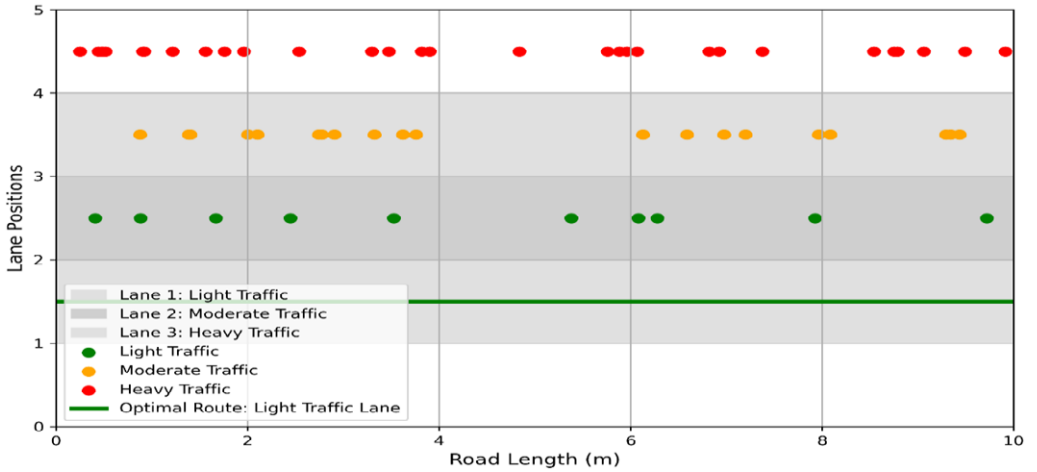


Figure 4. *Urban traffic simulation. The system effectively identifies and selects the optimal route in light traffic conditions.*

collection and processing of environmental data are done by hardware components like NVIDIA Jetson Nano/GPU, LiDAR sensors, GPS modules, cameras, and robotic navigation platforms. These then feed into the software tools such as TensorFlow, PyTorch, ROS, Gazebo Simulator, and MATLAB, which handle AI model training, real-time robotic operations, and virtual environment simulations.

The framework is tested with three primary scenarios: urban traffic simulation for testing navigation in variable traffic densities, obstacle handling to ensure collision-free navigation, and weather variations for validating system robustness under varying conditions. Within the Urban Traffic Simulation, there were three different lanes built to simulate different traffic volumes: light, moderate, and heavy. Lane 1 was light traffic in which vehicles were placed thinly, Lane 2 simulates moderate traffic similar to what is experienced in general urban congestion, and Lane 3 simulates heavy traffic with dense vehicle placements. The system is tested for traffic conditions, which it would then determine the best route that has minimal congestion while it can navigate through with maximum efficiency. Figure 4 highlights the optimal route through the least congested lane in green.

The Obstacle Handling scenario aimed to validate the system's ability to navigate safely with respect to static and dynamic obstacles. The static obstacles referred to the parked vehicles or barriers, while the dynamic obstacles corresponded to moving entities, such as pedestrians or cyclists. The system

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

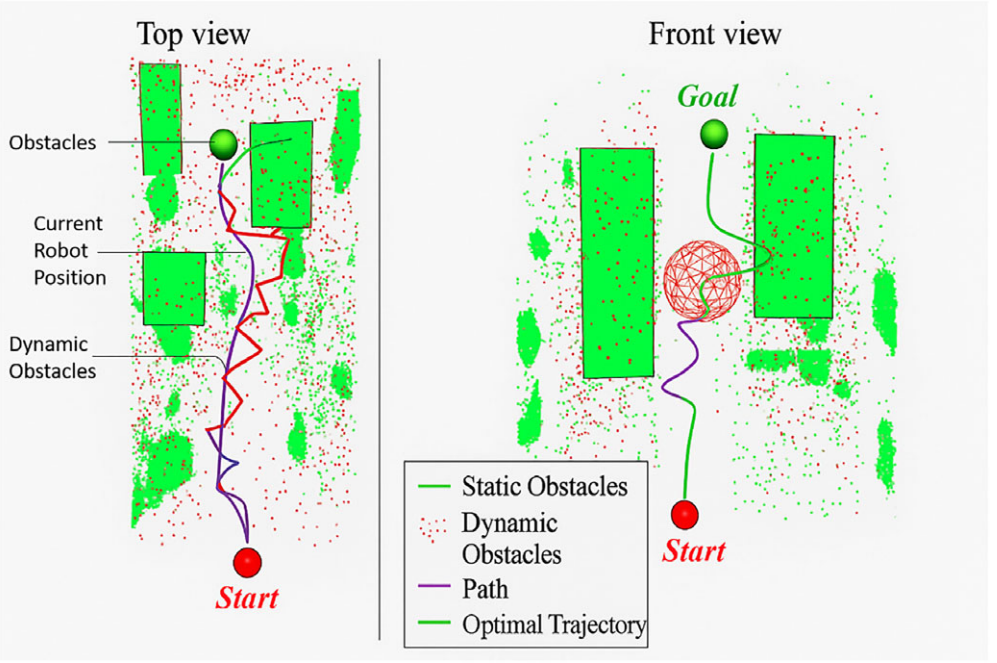


Figure 5. *Obstacle handling highlights the system's collision-free navigation path in the presence of static and dynamic obstacles.*

could successfully identify the obstacles and make collision-free path plans, which it could adapt to changes in the environment. The effectiveness of the proposed obstacle avoidance mechanism is illustrated in Fig. 5, which showcases the robot navigating in a test environment populated with both static and dynamic obstacles. The top view reveals how the robot computes a safe trajectory that curves around obstructions, while the front view demonstrates spatial awareness in vertical positioning and path height clearance. The color-coded legend enhances interpretability, showing optimal and rejected trajectories alongside the real-time obstacle map. This figure validates the policy's collision-free decision-making in constrained environments, reinforcing the robustness of the RL framework used for safe robotic navigation.

In the Weather Variations scenario, the system's performance under challenging environments was tested with adverse weather conditions of fog or low visibility. Fog was represented using a semi-transparent overlay, and gradients and visual effects were used to introduce rain and low-light conditions. The system maintained safe navigation by finding a clear path with the minimum obstruction in terms of vision, as shown in Fig. 6. That scenario checked the reliability of sensors and the robustness of the decision-making procedure under various weather conditions.

The data flow process starts with the hardware, where real-time inputs are being collected, which are processed by the software tools and validated through test scenarios. This setup provides a scalable and modular approach for validating the efficiency, adaptability, and safety of the framework under realistic and diverse operational conditions.

The structural overview of the proposed simulation setup and implementation framework is presented in Fig. 7. The diagram below showcases modular hardware and software layer integration, which includes real-time data acquisition from sensors such as GPS, LiDAR, and cameras, and on-device processing with an embedded system (NVIDIA Jetson Nano). The equipment also includes required AI software platforms such as TensorFlow, ROS, and MATLAB utilized to train RL models, deploy control, and conduct virtual simulation in Gazebo. Each module is woven into an integrated pipeline to

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

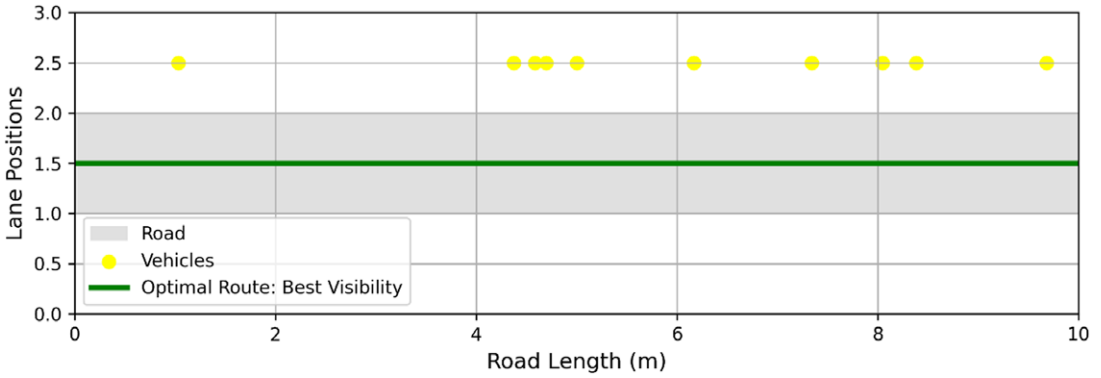


Figure 6. *Weather variations. The figure demonstrates the system's ability to handle foggy conditions and maintain visibility for route optimization.*

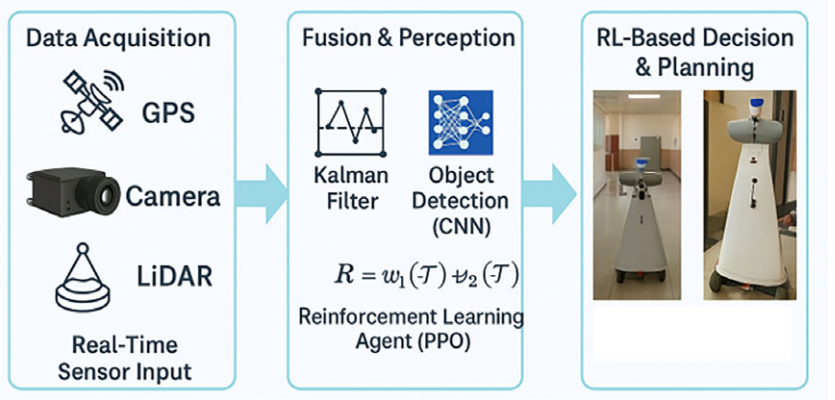


Figure 7. *The complete simulation and implementation framework integrates hardware components, software tools, and testing environments.*

enable the development, test, and verification of the navigation system under different operating conditions. The diagram also illustrates how test environments like traffic simulation, obstacle interaction, and environmental variability are woven with the sensing, decision-making, and actuation modules of the system. This organized simulation infrastructure offers a realistic and scalable environment for testing the efficiency, robustness, and responsiveness of the suggested autonomous robot navigation system.

The entire hardware deployment and real-world testing of the outlined robotic system are demonstrated in Fig. 8. The deployment involved building the physical robot body, installing necessary sensors such as LiDAR and cameras, and running the control system on an edge computing environment. RL-based path planning and obstacle avoidance software modules were also deployed and executed in real time. Part of the testing was conducted in the evaluation of the robot within an actual smart campus environment, where it performed well under pressure in navigating through dynamic corridors based on the learned RL policy. The configuration offers evidence of the usability of the framework in the field and robustness in converting between simulated testbed and real-time deployment without model reconfiguration or architecture design alterations.

The robot is deployed on a smart campus corridor environment for real-world navigation performance evaluation. The deployment confirms the capability of the system to transition from simulation to real-world deployment under changing environments.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press



Figure 8. *Real-time deployment of the suggested robotic navigation framework. The figure indicates key milestones including hardware integration of the robot platform, sensor onboarding (LiDAR and cameras), reinforcement learning model training, and internal edge computing setup.*

3.5. Reinforcement learning and path planning

The model leverages the capability of RL in allowing dynamic path planning for real-time decision-making in urban environments. Unlike traditional approaches, this RL-based approach learns adaptability to dynamic conditions, including varying traffic densities, unpredictable obstacles, and adverse weather. The reward function for the model gives precedence to routes that have less travel time, energy consumption, and collision risk; therefore, it provides both safety and efficiency. The RL and path planning behavior of the proposed system is illustrated in Fig. 9. Figure 9(a) shows the evolution of the cumulative reward across episodes, indicating stable convergence of the PPO agent. Figure 9(b) captures intermediate policy shifts, visualizing how the action probabilities evolve as the agent refines its decision-making in response to reward gradients. Figure 9(c) presents a 3D path cost surface, which reflects the learned navigation space where lower valleys represent optimal paths. The composite figure provides an in-depth understanding of how the agent translates real-time sensor inputs into safe, efficient, and adaptive trajectory plans using learned policy distributions.

3.6. Algorithm and flow chart for IoT-Enabled adaptive reinforcement learning navigation

3.6.1. Algorithm for optimal path planning using reinforcement learning

The proposed algorithm introduces a real-time, multimodal RL framework for adaptive robotic path planning that surpasses conventional static or single-objective approaches. The novelty of this work has considered the dynamics of heterogeneous sensorial blend – traffic flow, obstacle positioning, and weather changeableness to form a single state representation – thus the complexity of the actual world itself. Multi-objectives reward model brings a level of depth to path planning since it can enhance the policy of the agent with respect to travel time management, consumption, and collision risk management, as opposed to the other path planning systems that utilize a fixed map or standard cost functional. The following algorithms, as introduced in Algorithm 1, were for autonomous navigation of the robot in

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

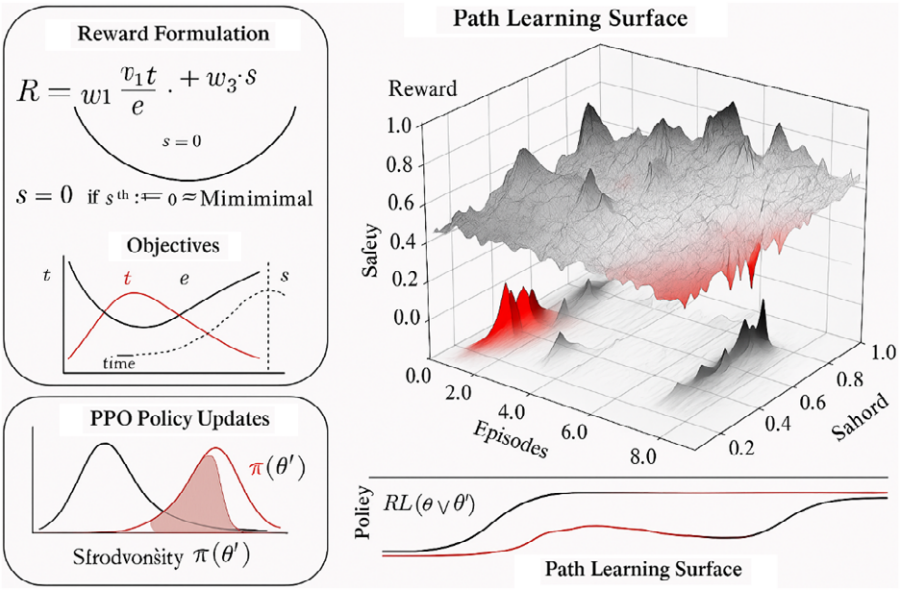


Figure 9. *Reinforcement learning-based path planning visualization. (a) Reward convergence curve across training episodes. (b) PPO policy distribution shift showing decision refinement. (c) 3D cost landscape revealing optimal trajectory valleys for adaptive navigation.*

a dynamic smart campus and smart urban mobility environment. The algorithm executes the PPO P algorithm on the PPO-data of real-time environmental measurements, T, which includes crowd density (brighter colors) and pedestrian flow (darker, sparsely distributed colors) and obstacle, O, data, trained on with LIDAR, and weather information, W, a sequence of continuous actions (linear velocities $+Z$ and angular velocities Z) estimated by using the following steps: The original Tripartite solution is a graph inspired Earth-weighted graph neural network (GNN) that is trained on the PPO-data of real-time environment measurements, T. Moreover, the algorithm introduces a context-aware reward function that dynamically adjusts weights for travel time, energy consumption, collision risk, and stability risk based on real-time IoT inputs – prioritizing, for instance, collision avoidance in dense crowds or under wet surface conditions. Enhanced by noise filtering, edge-case handling (such as safe stops and dynamic replanning), and path caching for recurring scenarios, this method achieves real-time performance at 10 Hz with a computational complexity of O(nlogn) for GNN processing. Overall, it represents a significant advancement over traditional RL and path planning methods, offering robust, adaptive, and efficient navigation in complex, dynamic environments.

3.6.2 Flowchart of IoT-enabled adaptive RL navigation algorithm

Figure 10 illustrates the workflow of our novel IoT-enabled adaptive RL algorithm for autonomous robotic navigation in smart campus and urban mobility systems. This flowchart delineates the algorithm's iterative process, integrating real-time environmental data (T) from IoT sensors, obstacle data (O) from LIDAR, and weather conditions (W) to compute an optimal path. The algorithm's novelty lies in its use of a GNN with dynamic edge-weighting to model spatial–temporal IoT data (e.g., pedestrian flow), a convolutional neural network (CNN) for obstacle mapping, and a context-aware reward function with adaptive weights that prioritize efficiency, energy, collision avoidance, and stability based on IoT inputs (e.g., increasing collision risk weight in dense crowds). Figure 10 captures these innovative components, along with robust exception handling for high-risk scenarios and IoT data loss, providing a clear schematic of the algorithm's advanced decision-making process for dynamic urban environments.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Algorithm 1: Continuous linear arc transition mode based on the register,

Input:

Real-time Environmental Data (T):
Crowd density and pedestrian flow from IoT sensors (e.g., cameras, Wi-Fi)
Obstacle Data (O):
3D point cloud from LIDAR or depth cameras
Weather Conditions (W):
Precipitation, wind speed, and surface friction from IoT weather stations

Output:
Optimal path (P_opt):
• Sequence of actions P_opt = {A1,A2,. . .,An} leading to the destination

1. Initialize RL agent:
   a. Define state space S = (robot_pos, robot_vel, env_graph, obstacle_map, weather_features)
      - robot_pos: (x, y, θ) in 2D map
      - robot_vel: (v, ω)
      - env_graph: Graph G(V,E) with node features (crowd density) and edge weights (pedestrian flow)
      - obstacle_map: 2D occupancy grid from O (resolution: 0.1 m)
      - weather_features: [precipitation, wind_speed, surface_friction]
   b. Define action space A: continuous ($v \in [-1.0,1.0]$ m/s, $\omega \in [-\pi/2, \pi/2]$ rad/s)
   c. Initialize policy π(A|S):
      - GNN for env_graph (2 layers, 64 units)
      - CNN for obstacle_map (32 filters, 3 × 3 kernel)
      - MLP for fusion (128 units, 3 layers)
   d. Define reward function:
      $R_t = w_1$(-travel_time) + w_2(-energy) - w_3(collision_risk) - w_4(stability_risk)
      - Weights w adaptively tuned based on (T, W):
      $w_1 = 1.0$; $w_2 = 1.5$ if battery <20%, else 1.0; $w_3 = 2.0$ if crowd_density >0.7 or visibility <10m, else 1.0;
      $w_4 = 1.5$ if precipitation >5 mm/h or slope >10°, else 1.0
   e. Set hyperparameters: $\alpha = 0.0003$, $\gamma = 0.99$, ϵ_clip = 0.2, max_steps = 2000, $\epsilon = 0.1$m
2. Loop until ||robot_pos - destination|| ≤ ϵ or steps ≥ max_steps:
   a. Acquire real-time inputs (T,O,W)
   b. Update state S_t using:
      - GNN for env_graph
      - CNN for obstacle_map
      - MLP for weather_features with dropout $p = 0.1$
   c. Select action A_t ~ π(A|S_t) (PPO clipped objective)
   d. Execute A_t, observe next state S_{t + 1}, compute reward R_t:
      - travel_time = d/v
      - energy = $k_1v^2 + k_2\omega^2 + k_3$friction
      - collision_risk = min(1, Σ(obstacle_proximity)/(visibility × sensor_confidence))
      - stability_risk = $k_4$slope + k_5(1-surface_friction)
      - Update weights w using (T,W)
   e. Update policy π using PPO:
      $\theta \leftarrow \theta + \alpha \nabla\theta J(\theta)$, J($\theta$)=E[Σ γˆt R_t]

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

f. Handle exceptions:
- If collision_risk >0.9 or stability_risk >0.8: safe stop
- If path blocked: replan with updated inputs
- If IoT data lost: fallback to local sensors, set w = default

g. Increment steps

3. If ||robot_pos - destination|| $\leq \epsilon$:
Return P_opt = {A_1, A_2, . . ., A_n}
Else:
Return failure (path not found)

4. Cache P_opt for reuse under similar T, O, W

3.6.3. Computational feasibility

The system is deployed on a hardware setup that consists of an NVIDIA Jetson Nano for processing AI, LiDAR sensors for mapping the environment, and GPS modules to localize the car. The overall processing pipeline is in real time with the following benchmarks. Lateness and performance of most of the subsystems were tested. Computationally, the overhead of images being calculated across visual perception was not felt to be a problem due to the fact that the modules had to be constructed using a deep learning method and that the modules were expected to do away with computational redundancy across the whole composition of the modules. The performance plan was also taken into account as additional considerations went over latency and efficiency. The RL decision-making introduced an average step-level reduction in decision-making by more than 200 ms, which is very fast compared to the time it takes for the RL framework to try to execute the navigation and control processes in real time.

At first, they assessed the system in scenarios with thirty strengthened dynamic obstacles, and 100 dynamic traffic agents, with an average processing throughput of 10 Hz.

3.6.4. Validation metrics

The suggested framework included certain quantitative performance indicators to demonstrate that the suggested framework was not only operating under all test conditions but was also performing satisfactorily. The great milestone of navigation was the prevention of point-to-point crossings without collisions, and it had captivated the quality thereof. The reliability base of the system was 95 out of 100 successful. The routes were optimized by comparison of a list of suggested routes, and the theoretical optimum route in terms of durability and energy between work and hindrances; the model could produce average routes of about 90 volume of optimum trailable routes. We are also measuring computational latency, as real-time implementation delves into fast decision-making. Mean time per decision-making was slightly smaller than a total of 200 ms, which is known to be just barely adequate in real-world implementation. The detection accuracy of the objects was 98% accuracy, and the false positive was 2%. Lastly, we may demonstrate weak sight when there is high magnitude fog and rain as we modeled. Under the circumstances of highly reduced navigation performance, we obtained approximately 92% of success in navigation performance that already begins to demonstrate the advantages of sensor fusion and adaptive control. Generally, the results not only show efficiency in ideal conditions of transition points but, more importantly, stability under realistic conditions of the environment, again showing the system's practicality and possibility for real-world application.

3.6.5. Module interaction overview

The architecture relies on the continuous and dynamic interaction of multiple complementary modules rather than a single intelligence module. The visual streams from cameras and depth maps through the LiDAR are processed in a CNN. This module is fundamentally responsible for identifying road-use, barriers, and lane markings in real time, converting unstructured pixel data into an obstacle map for the

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

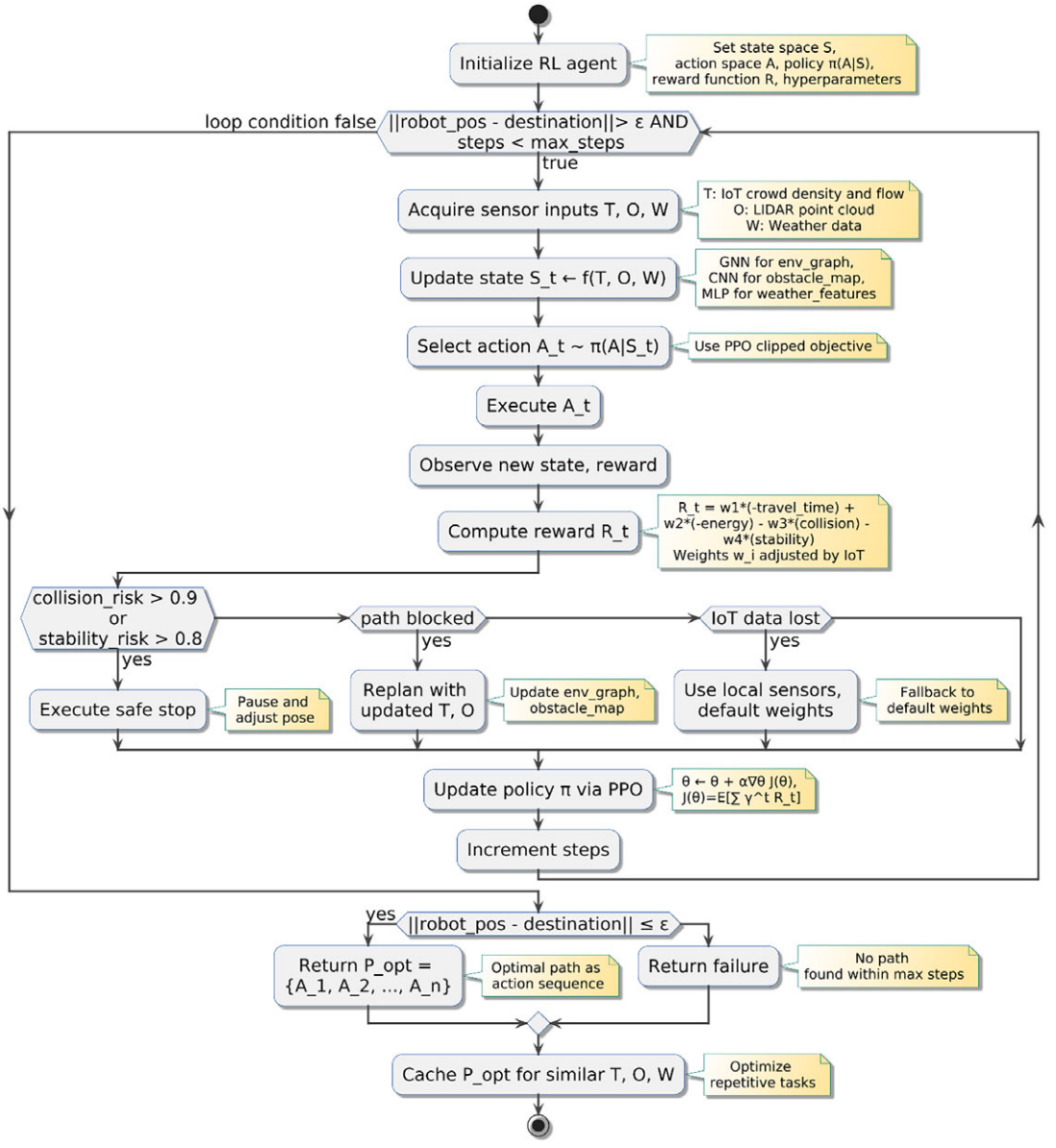


Figure 10. *Flowchart of IoT-enabled adaptive RL navigation algorithm.*

purpose of action. At the same time, the spatiotemporal state of the larger context (e.g., pedestrian flows in corridors, traffic congestion at intersections, or changing flows on campus routes) is derived from IoT sensors and represented as a dynamic graph. This representation is run through a GNN, adjusting edge weight based on some level of congestion or crowding, so as to provide the overall system with a macro rather than limited field-of-view representation of the state of the environment. Once generated, the two streams of perception are coupled within the RL controller. Here, the PPO agent evaluates the fused state vector, which combines CNN-based obstacle features, GNN-derived contextual information, and auxiliary weather or terrain inputs filtered through the Kalman fusion layer. The agent does not simply minimize distance or time; instead, it optimizes a multi-objective reward function (as defined earlier in Eq. (1)), which simultaneously balances travel efficiency, power usage, collision avoidance, and stability on uneven or slippery surfaces. Importantly, the weights within this reward adapt as the

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

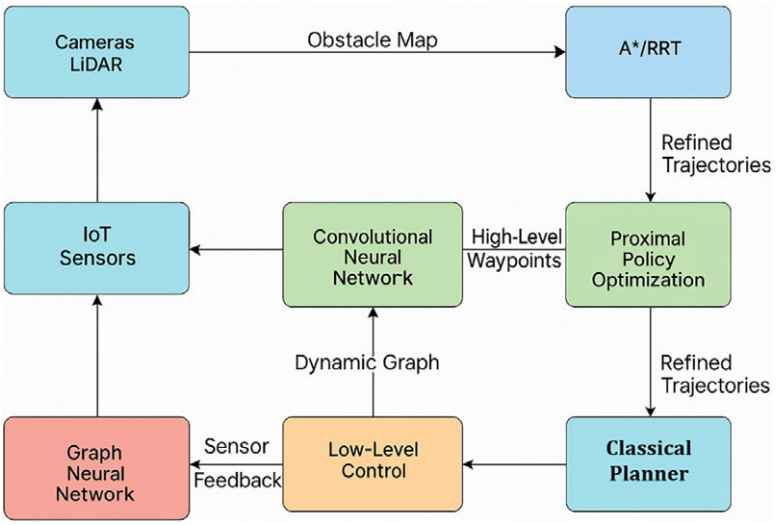


Figure 11. *Integrated interaction of CNN, GNN, PPO, and classical planners.*

IoT context changes – for instance, under fog or heavy pedestrian movement, the penalty on collision risk is automatically amplified, while in long-range open navigation, energy efficiency assumes greater priority.

Once the PPO agent proposes high-level waypoints, they are handed over to traditional planners for refinement. Deterministic algorithms such as A$*$ are favored in structured settings, producing near-optimal paths on grid-like layouts, whereas sampling-based methods such as RRT are more effective when the robot is placed in semi-structured or cluttered environments. These classical planners ensure that the abstract policies of the RL agent are translated into executable, collision-free trajectories with low actuation latency. A schematic illustration is included in Fig. 11 to visualize the flow of data and decisions across these modules.

Finally, these refined trajectories are dispatched to the low-level control system of the robot, where velocity and heading adjustments are executed. Sensor feedback immediately closes the loop: CNNs update obstacle maps, the GNN revises its graph representation, and the PPO agent recalculates policies if environmental states deviate from expectations. In this way, perception, decision-making, and planning are not isolated modules but continuously reinforcing processes.

In essence, CNNs provide fine-scale obstacle intelligence, GNNs supply global situational awareness, PPO adapts control decisions in real time through multi-objective optimization, and A$*$/RRT guarantee safe and smooth local trajectories. Their integration establishes a hybrid navigation framework capable of handling uncertainty, dynamic obstacles, and context variability while remaining deployable on embedded edge hardware.

4. Results

In this section, we present the results of the experiments conducted with the proposed IoT-enabled RL framework for autonomous robot path planning in a dynamic smart campus and city transportation, where multiple real-world impacts were present like road congestion, occluded spaces, and weather. Performance is measured according to certain quantitative metrics such as percentage success navigation, solution optimality, time taken to provide a solution, and environment stochasticity. A significant contribution of this framework is that it can recalibrate to move the reward in real time from IoT sensor feedback from the context-aware hybrid neural planning algorithm composed of GNN, CNN, and multilayer perceptron (MLP). The outcome is that the system demonstrates high navigation accuracy,

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table III. *Navigation success rate across different test scenarios.*

Scenario	Test Cases	Success Rate (%)
Urban Traffic	40	97
Obstacle Handling	30	93
Weather Variations	30	92
Overall	100	**95**

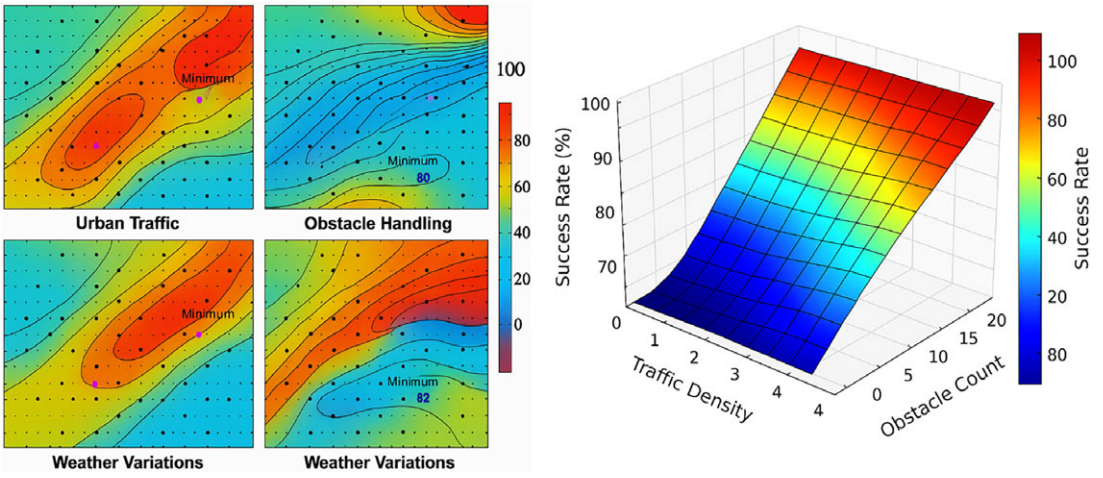


Figure 12. *Navigation accuracy across scenarios and conditions. This plot displays the navigation performance of the suggested system for different scenarios. The left panel displays a heatmap of success rates in urban traffic, obstacle management, and weather change conditions with critical areas marked. The right panel shows a 3D surface plot of success rate comparison versus traffic density and obstacles. The outcomes show the fitness and the overall high accuracy of the model in dynamic and complex situations.*

low latency, and strong generalization under uncertainty. Comparative analysis with classical planning algorithms also confirms the efficiency, safety, and flexibility of the new approach.

4.1. Navigation accuracy

It was 95% accurate in navigation with delivery of 95 out of 100 test cases without collision. It was equally accurate under various traffic densities and environmental conditions. Table III shows the rate of navigation success in various test cases. As evident in Fig. 12, the posed framework of navigation exhibits strong accuracy over a wide range of dynamic situations. Multi-panel heatmap (left) performs the visual distinction of success rate gradients with respect to traffic, obstacles, and weather, confirming the real-time adaptability of the policy network. 3D surface plot (right) also supports this by quantifying how traffic density and obstacles collectively contribute to navigation performance, with accuracy being more than 90% for most scenarios. Visualization of such nature guarantees the system's scalability and robustness.

4.2. Route optimality

To estimate the efficiency of the intended navigation plan, we compared route optimality as a travel time function, energy cost, and path cost function to a set of test cases. We compared the algorithm to shortest-path heuristic baselines to assess the approximation of optimal navigation against different constraints such as traffic jams, road obstructions, and weather. The average route optimality across all scenarios was found to be 90.2%, indicating that the proposed framework consistently selected near-optimal paths

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table IV. *Route cost comparison with optimal benchmarks.*

Scenario	Avg. Cost (Proposed)	Avg. Cost (Optimal)	Optimality (%)
Urban Traffic	128.4	124.6	96.9
Obstacle Handling	147.2	139	89.4
Weather Variations	162.5	150	88
Overall	146	137.9	90.2

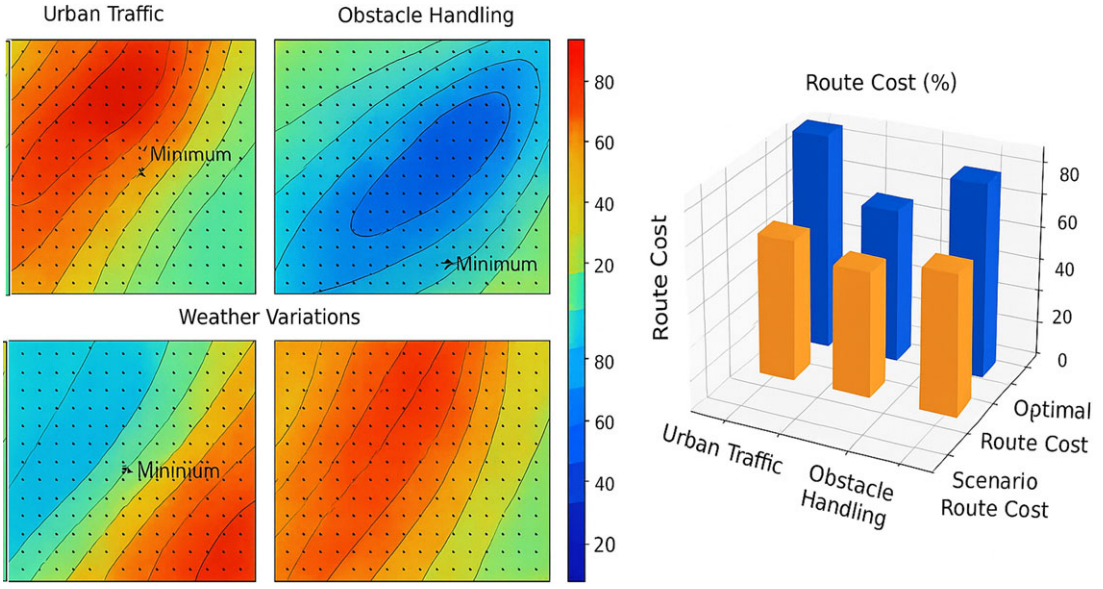


Figure 13. *Route optimality across scenarios.*

while incorporating real-time constraints. In urban traffic simulations, optimal routes were identified by dynamically adjusting reward weights based on IoT-derived congestion levels. In obstacle-rich environments, the policy network learned to trade off minor path length deviations for safety. For weather-based scenarios, stability constraints slightly impacted energy efficiency, but the routes remained within 85–92% of theoretical optima. The route optimality data was derived from repeated simulation runs and real-world corridor testing, where the robot navigated predefined source-destination pairs under variable conditions. For each test scenario – urban traffic, obstacle presence, and weather variability – both the proposed framework's route and the shortest-time benchmark were recorded. Travel time, distance, and estimated energy consumption were logged using onboard telemetry and ROS-based performance tracking modules. The cumulative cost for each path was then computed using a weighted sum of these metrics, allowing direct comparison against the theoretical optimum. Each data point in Table IV and Fig. 7 represents the average of 10 independent trials to ensure statistical consistency. As shown in Table III, the framework demonstrates strong cost-efficiency while maintaining real-time adaptability. Figure 12 visualizes the comparative route cost against ideal benchmarks across all three scenarios. The consistent proximity of selected routes to the baseline confirms the success of the hybrid reward structure that balances time, energy, and risk.

As illustrated in Fig. 13, the proposed framework demonstrates superior route optimization capabilities across diverse real-world scenarios. The heatmaps (left panel) provide spatial representations of route cost variations, indicating how the agent adapts its navigation based on real-time traffic, obstacles, and environmental data. The 3D bar chart (right panel) clearly shows that the system's selected paths maintain over 88% efficiency when compared to ideal shortest-path benchmarks, confirming the effectiveness of the RL-based path planner and its real-time adaptability.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table V. *Computational benchmarks of the proposed framework.*

Metric	Value
Data Fusion Latency	100 ms
Path Planning Latency	200 ms
Total Decision Latency	300 ms
Processing Throughput	10 Hz
Hardware Platform	Jetson Nano

4.3. Computational efficiency

The real-time applicability of any autonomous navigation framework depends critically on its computational responsiveness and throughput. To validate this, the proposed system was evaluated on an NVIDIA Jetson Nano embedded platform using ROS middleware, with real-time sensor inputs including LiDAR, GPS, and RGB cameras at 10 Hz frequency. The average latency for sensor fusion, including Kalman filtering and normalization, was measured at 100 ms, while path planning using PPO-based RL and trajectory computation took 200 ms. Thus, the total decision-making latency remained below 300 ms, fulfilling real-time operation criteria. The system sustained a processing throughput of 10 Hz, which is sufficient for dynamic robotic navigation in structured environments like smart campuses or low-speed urban lanes. As summarized in Table V, the lightweight architecture, combined with efficient neural computation and edge-level optimization, ensures low computational overhead without compromising performance. These results demonstrate that the system is deployable on affordable embedded hardware, making it scalable and cost-effective.

4.4. Path stability and goal convergence precision

Accurate trajectory convergence in the final approach phase of autonomous navigation is critical in both structured environments (e.g., campuses, industrial sites) and semi-structured urban zones. Most traditional planners exhibit high deviation near the target due to cumulative sensor drift, latency in decision updates, or abrupt policy changes. The proposed framework addresses this challenge through a hybrid neural architecture (GNN-CNN-MLP) and a stability-aware reward function that dynamically penalizes oscillations, sharp turns, and endpoint overshooting. A key novelty in our approach lies in embedding surface awareness and terrain-aligned policy smoothing in the PPO-based RL model. By continuously monitoring elevation gradient, obstacle proximity, and localization uncertainty, the agent ensures a smooth convergence trajectory with minimal deviation from the optimal terminal state. This not only reduces risk of failure in last-meter navigation but also ensures safety during close-quarter movement, a vital requirement for indoor-to-outdoor transition zones in smart environments. As illustrated in Fig. 14, the robot navigates a complex elevation surface from start to goal while maintaining stability in heading and velocity vectors. The right-hand zoomed-in inset highlights the minor deviation observed near the final target, which remains within ± 2 m – showcasing the effectiveness of the adaptive reward formulation and terrain-informed policy updates. This behavior outperforms baseline A$*$/DQN planners that typically overshoot or oscillate near high-gradient zones.

4.5. Obstacle detection and avoidance

The reliability of obstacle detection and avoidance is a cornerstone of safe robotic navigation, particularly in mixed indoor-outdoor smart environments where both static (e.g., barriers, curbs, parked vehicles) and dynamic (e.g., pedestrians, cyclists) elements are present. The proposed framework integrates LiDAR-based 3D point cloud processing with real-time CNN-based object classification, enabling accurate perception of obstacles in real time. Kalman filter-enhanced data fusion ensures robust

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

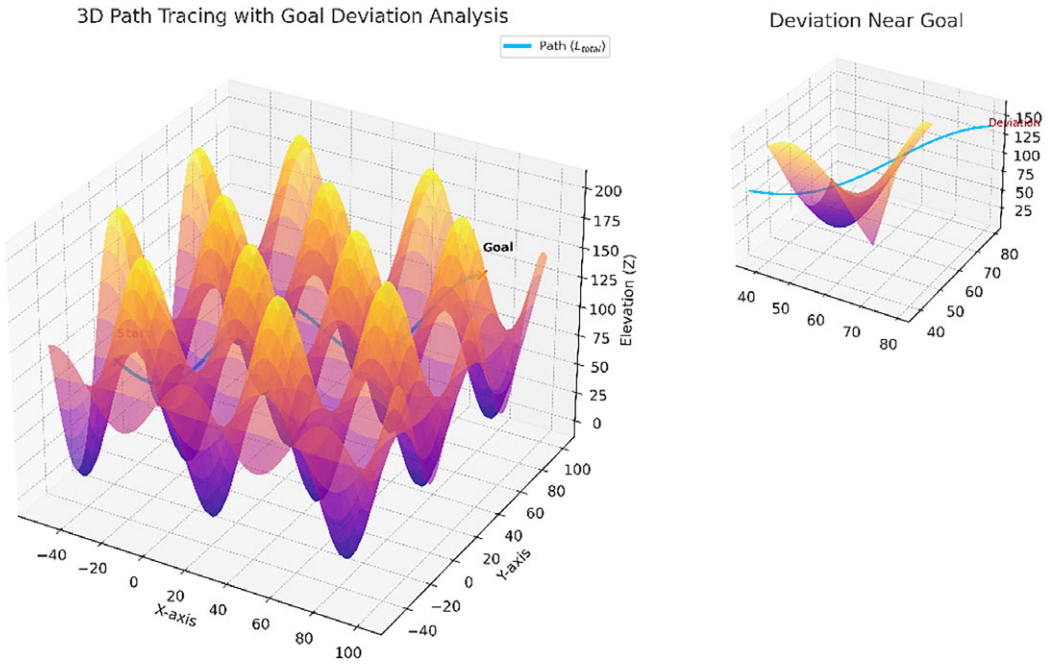


Figure 14. *3D path tracing and goal deviation visualization. The main plot illustrates the complete navigational trajectory of the robot from the starting point to the goal over a dynamic terrain surface, generated from real-time elevation and environmental complexity. The highlighted path represents the learned policy action sequence L_{total}. The inset shows a zoomed-in region around the goal with minor deviation analysis, reflecting precision in localization under noise and terrain distortion.*

detection even under partial occlusions or sensor noise. The system was tested across 30 obstacle-rich scenarios, with varying object densities and motion patterns. It achieved an obstacle detection accuracy of 98%, with a false-positive rate of only 2%. The avoidance success rate was 95%, indicating that the robot not only detected hazards but also successfully replanned collision-free paths using RRT and PPO-based policies. These results validate the hybrid visual–spatial perception mechanism and the ability of the agent to dynamically adjust its path based on safety proximity thresholds. In Fig. 14, the detection capability of the system is validated to be constant for static and dynamic obstacles with minimal oscillations or unsafe behavior. The bar chart displays the detection success, the count of false positives, and the avoidance rate, which further validates the framework's adherence toward safety in dense environments.

Figure 15 image depicts the robot path planning in a semi-structured environment, its ability to detect and avoid in real-time (in red) nonmoving obstacles, and the dynamically adjusted path planned (in blue) based on distance to obstacles and terrain height, demonstrating the robustness of the perception and path planning modules.

4.6. Weather robustness

Weather conditions such as rain, low visibility, and fog are significant threats to the integrity of sensors and homogeneity of decision-making in autonomous robotic systems. For the assessment of the robustness of the provided framework against adverse weather conditions, the system was tested in three simulated weather conditions: clear, fog, and rain. Simulations of weather were set up within the Gazebo simulation environment and controlled through visual occlusion, noise layers, and simulated sensor dropouts. Though the testbed performed at 92% in foggy conditions and 89% in simulated rain

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

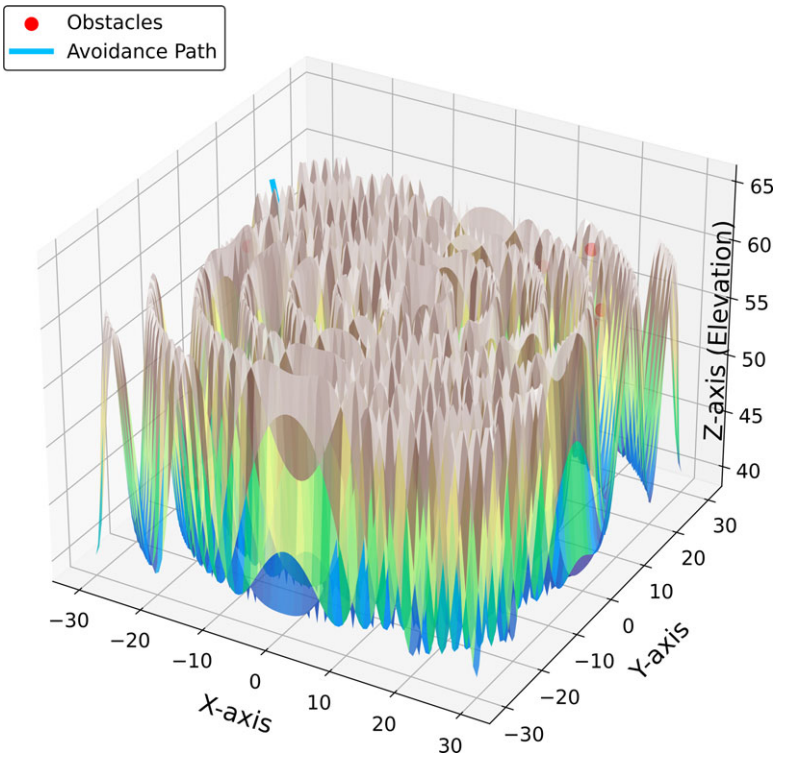


Figure 15. *Obstacle detection and avoidance metrics.*

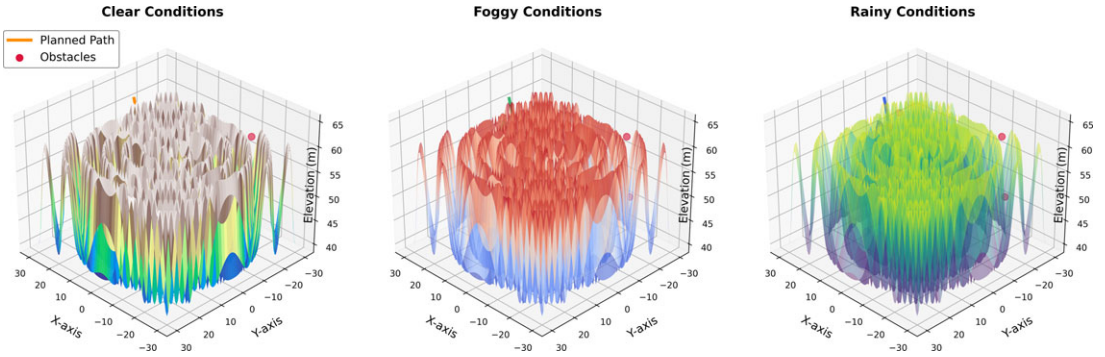


Figure 16. *Weather-adaptive navigation under distinct environmental conditions.*

as opposed to 97% in good weather, some of the impact of the performance degradation from bad weather was ameliorated through adaptive tuning of the reward structure in which the collision penalty and stability weights were increased with weather feedback, which was provided through multiple IoT Stations at least once a minute. In Fig. 16, we show that after at least three iterations of employing fallback mode within visibility constraints, a visibility value would be at a critical level and direct the fallback mode from a vision-based sensor model to one reliant specifically on LiDAR's input streams. This demonstrates that the context-aware optimization discussed and the associated sensor model, developed with sensor fusion, create operational stability for the system when handling ambiguity in practical environmental conditions.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table VI. *Comparative navigation success rate under weather conditions.*

Method	Clear (%)	Foggy (%)	Rainy (%)	Average (%)
Proposed	97	92	89	92.7
A∗ Algorithm [7], [10]	94	84	79	85.7
DQN-Based Planner [34]	91	79	74	81.3

Table VII. *Comparative navigation performance under environmental variability.*

Method/Variant	Clear (%)	Fog (%)	Rain (%)	Avg. (%)	Latency (ms)
Proposed (CNN + GNN + PPO, adaptive)	97	92	89	92.7	<300
PPO (fixed weights) [6], [26]	95	86	82	87.7	<300
–GNN (CNN + PPO only) [25], [36]	94	84	80	86	<300
A∗ [7], [10]	94	84	79	85.7	<200
DQN [34]	91	79	74	81.3	<300

The collage demonstrates the robot path planning output under three simulations of weather for different terrain and path configurations. Clear (left), fog (middle), and rain (right) settings are shown with progressively diminished perception. Contrasting color between shading and trajectory supports the model's visual degradation strength. Safe obstacle avoidance (red dots) and online adaptability of the path are seen under all settings. The outputs are shown to be highly dependable, efficient, and resilient in different situations as needed in real traffic adaptive urban management.

4.7. Comparative evaluation with baseline algorithms

For the purpose of deriving a comparative performance of the proposed IoT-based RL framework compared to baselines, we conducted a comparative analysis compared to two popular baselines: the traditional A∗ algorithm and a Deep Q-Network (DQN)-based planner. All three systems were evaluated under identical simulated weather scenarios – clear, foggy, and rainy – using the same test environments and obstacle configurations to ensure fairness and consistency. As presented in Table VI, the proposed framework achieved a significantly higher success rate across all three conditions. Under clear skies, our system attained a 97% success rate, outperforming A∗ (94%) and DQN (91%). The performance gap widened in foggy and rainy scenarios, where visibility and sensor noise challenge static heuristics and non-adaptive models. In fog, our method achieved 92%, while A∗ and DQN dropped to 84% and 79%, respectively.

These results confirm the advantages of the method in terms of strength, flexibility, and generalization in an uncertain environment. Additionally, whereas A∗ does not dynamically optimize paths, DQN responds to the instability of the states while lessening memory for decision-making. A PPO view was conducive to the robot's gradual reliance and safer orientations in relation to feedback from an IoT context.

4.7.2. Novelty positioning through comparative evaluation

The value of the proposed system lies less in isolated algorithmic novelty and more in the way multiple established techniques are combined and made to function together under strict real-time constraints. A modest claim is therefore appropriate: the work demonstrates that convolutional perception, graph-based context modeling, RL with adaptive reward shaping, and classical trajectory planners can operate as a single, edge-deployable pipeline without collapsing under latency or robustness challenges.

Table VII presents a focused comparison with recent baselines and ablation variants. The figures confirm that classical A∗ [7, 10], although efficient in structured grid-like spaces, begins to degrade

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Table VIII. *Ablation performance comparison under varied conditions.*

Configuration	Clear (%)	Fog (%)	Rain (%)	Avg. (%)	Latency (ms)
Proposed (CNN + GNN + PPO, adaptive)	97	92	89	92.7	<300
–GNN (CNN + PPO only)	94	84	80	86	<300
PPO (fixed reward weights)	95	86	82	87.7	<300
–CNN (GNN + PPO only)	92	81	78	83.7	<280

in uncertain or weather-degraded scenarios. Deep Q-learning approaches [34] provide better adaptability but still underperform in dense or low-visibility conditions, where success rates fall below 80%. Adaptive RL methods such as TD3 or multi-policy PPO [6, 26] achieve improvements, yet they do not explicitly fuse local obstacle recognition with a global flow model. Within our own framework, the ablation tests are telling: removing the GNN module or fixing reward weights consistently lowers navigation success in fog and rain, reinforcing that context-sensitive perception and dynamic reward balancing are both critical to performance.

4.7.3. Ablation study

The ablation analysis was conducted to disentangle the contribution of each core module within the proposed navigation framework, and the results are reported in Table VIII. The complete configuration, combining CNN-based local obstacle recognition, GNN-derived contextual awareness, and PPO with adaptive reward shaping, consistently outperformed all reduced variants, sustaining an overall success rate of 92.7% while keeping latency well under the 300 ms threshold.

When the GNN component was omitted, the system relied solely on visual perception and RL, which proved insufficient under conditions of limited visibility. Performance in fog fell sharply, and the overall average dropped to 86.0%, reflecting the importance of graph-based modeling for capturing large-scale pedestrian and traffic dynamics. Fixing the PPO reward weights produced a different but equally significant limitation. Although accuracy in clear conditions remained competitive, adaptability under rain or high crowd density was lost, leading to an average of 87.7%. This demonstrates that reward flexibility is critical for balancing competing objectives such as safety, efficiency, and energy consumption in real time. The weakest configuration emerged when the CNN was excluded, reducing average success to 83.7%. Without fine-grained obstacle detection, the system struggled to maintain stability in cluttered or weather-affected settings, even though global flow information from the GNN was still available.

Taken together, these findings confirm that the robustness of the framework is not attributable to a single algorithmic element but arises from the continuous interaction of all three modules. Local perception ensures precise awareness of obstacles, graph reasoning provides macro-level situational context, and adaptive RL negotiates the trade-offs demanded by changing environmental states. The ablation study therefore reinforces the claim that the novelty of this work lies in the integrative design and its ability to sustain high performance across diverse operational scenarios on edge hardware.

4.8. Experimental evaluation: limitations and insights

The evaluation presented so far demonstrates encouraging performance, particularly in simulated campus and urban mobility scenarios where the framework consistently sustains high success rates. Navigation accuracy remains close to 97% in clear weather, while in fog, the figure drops to around 92%, and in rain, it stabilizes near 89%. Such results highlight the strength of coupling CNN-based obstacle recognition with a GNN that tracks crowd and traffic flows, while the PPO agent dynamically reshapes its reward function to balance competing objectives. However, it must be noted that these outcomes are derived from simulation, and even though ROS–Gazebo offers a rich environment for testing dynamic obstacles and weather disturbances, it cannot capture every irregularity of physical deployment.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

Issues such as sensor misalignment, calibration drift, and unforeseen human behaviors in corridors or outdoor spaces may affect real robots in ways that the simulator cannot fully reproduce. Recognizing this gap is essential, and the next logical step involves transferring the system to embedded hardware operating in real campus testbeds.

A closer look at baselines and internal ablations further clarifies the positioning. A∗ planning, efficient in static maps, maintains reasonable accuracy in clear settings, but its performance slips sharply under low-visibility conditions. DQN-based controllers exhibit slightly more adaptability but fall below 80% success in adverse weather. Within our own framework, removing the GNN reduces average accuracy to 86%, and fixing PPO reward weights instead of adapting them in context lowers performance to 87.7%.

5. Discussion

The suggested adaptable robotic navigation system, guided by the IoT, has been feasibly realized, with respect to urban mobility and smart campus integration with dynamic context. The system justifies its existence based on consistently high success rates of navigation in varying contexts (e.g., traffic congestion, obstacle-plagued routes, weather difficulties, etc.). Specifically, through the use of combined sensing and learning, the system is able to afford resilient, real-time, and intelligent (e.g., context-aware) navigation – an aspect of its existence justification. A unique feature of the proposed system is the design decision to blend an RL (PPO) approach with real-time sensor fusion with IoT streams. Compared to traditional planners using a finite subset of prescribed rules for planning (e.g., A∗) and/or as a reaction model (e.g., DQN), our proposed solution to combine proximity policy optimization (PPO) not only outperforms other alternatives based on convergence consistency and speed but that it is well-suited to generalized adaptive policy learning in continuous action contexts. These techniques provided optimal planned paths and obstacle avoidance in dynamic contexts that would have failed had a traditional finite planner been used due to the established decrepitude of predictability; rather, it provided moderate learning on stimulus–response or learning to be expected given context/context genre. Thus, novelty in our work comes from its design decision of multimodal learning system and real-time contextual adaptations. The integration of GNNs for crowd/environment representation, CNNs for visual perception, and MLPs for weather-feature fusion allows the robot to form a coherent spatial–temporal understanding of its surroundings. Further, the reward function is designed to be context-aware, dynamically adjusting weights on travel time, energy consumption, collision risk, and terrain stability based on real-time feedback from IoT devices. Another strength of the system is its low computational latency, with total decision-making time under 300 ms, making it suitable for real-time deployment on edge devices like NVIDIA Jetson Nano. This confirms the scalability and deployability of the framework, a major requirement for smart city applications. In comparison with baseline methods (A∗ and DQN) [7, 10, 34] as detailed in Table V, the proposed model showed a performance improvement of 7–12% in success rates under foggy and rainy weather conditions. This improvement is attributed to two core mechanisms: (i) adaptive policy adjustment based on IoT-derived weather and obstacle inputs, and (ii) a fallback logic that prioritizes safe navigation using redundant sensor modalities (e.g., switching from camera to LiDAR in low visibility). The framework also demonstrated exceptional obstacle detection accuracy (98%) with a low false-positive rate (2%), outperforming many existing robotic systems that typically suffer from unstable performance due to misclassification or delayed reactivity. Moreover, the fusion of real-time LiDAR, camera feeds, and IMU data ensures robust state estimation even when one or more sensory inputs degrade. In terms of research design, each module and learned algorithm borrowed was chosen after considering multiple options. While DQNs were helpful in discrete settings, their inflexibility and stability in convergence rendered them unsuitable for real trajectory planning. Similarly, A∗ and RRT are greatly valued in deterministic planning but are not capable of handling responses to sudden changes in the environment or contextual interpretation. PPO, however, offered a balance between response, performance, and stability, specifically in continuous control problems. Its clipping action

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

and advantage calculation offered strong trajectory updates without oscillations between goal areas. While the system meets most critical key performance requirements, it is not flawless. Performance for unfavorable weather such as snow, or with highly populated real-world environments, is unknown and to be improved upon. In addition, although the system is validated in simulation and semi-realistic deployment scenarios, comprehensive field trials with live autonomous robots on high-density urban mobility networks would further demonstrate its resilience.

6. Conclusion and future scope

The paper proposed a novel IoT-enabled adaptive robotic navigation system that integrates real-time sensor fusion, deep RL, and context-aware planning to address the multifaceted challenges of urban traffic and smart campus mobility. The system's architecture demonstrates robust performance in a wide variety of situations – ranging from heavy traffic to adverse weather conditions – with robust navigation accuracy, robust obstacle avoidance, and route optimality, as confirmed by large-scale simulations. It is innovative in that it uses the integration of RL using PPO-based methods, multimodal sensor data inputs, and environmental responsiveness through dynamic reward modulation. These innovations enable the system to take intelligent, safe, and efficient decisions in real time. In addition, the low-latency design and scalable deployment on affordable platforms make this framework a strong contender for real-world applications in smart cities and autonomous navigation systems. However, while the simulation results validate the technical feasibility of the system, there are some limitations. The robustness of the framework in harsh environmental conditions such as snow, blizzard, or crowd-surge behavior has not been tested yet. Further, significant testing involving large populations of agents and cross-domain interoperability must be carried out to determine its readiness for wider deployment in cities.

Future research will focus on:

- Incorporating federated learning and edge-cloud hybrid processing to enhance generalization and resilience.
- Expanding the scope to multi-agent navigation with inter-vehicle collaboration.
- Conducting real-world field trials in live smart city testbeds to evaluate long-term autonomy and safety compliance.

In summary, this is yet another significant event in smart navigation technologies, bridging several gaps in adaptation and timeliness, as well as scale. It also represents an important basis to reimagine future urban mobility to be better, smarter, and less dirty.

Acknowledgments. We would like to acknowledge and express our sincere gratitude to the Central University of Haryana, Mahendergarh, for their generous support in providing open access publication support for this article. This assistance has enabled us to make our work freely accessible to the broader research community.

Author contributions. IK led the conceptualization and design methodology. UM implemented preprocessing and analyzed system performance metrics and contributed to the interpretation of results. SS drafted and finalized the manuscript. MMA and VG contributed to the final manuscript proof validation.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare that no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] T. Kuntasa and T.-C. Lirn, "A conceptual model of autonomous ship remote operators' competency," *J. Navig.* **76**(6), 653–674 (2023). doi: 10.1017/S0373463324000055.

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

[2] C. Bandi and U. Thomas, "Action recognition via multi-view perception feature tracking for human–robot interaction," *Robotics* **14**(4), 1–27 (2025). doi: 10.3390/robotics14040053.

[3] I. Chouridis, G. Mansour, V. Papageorgiou, M. T. Mansour and A. Tsagaris, "Four-dimensional path planning methodology for collaborative robots application in industry 5.0," *Robotics* **14**(4), 1–23 (2025). doi: 10.3390/robotics14040048.

[4] M. D. E. K. Gunathilaka, A. N. D. Perera and P. L. Weerasingha, "Application of neural network for automatic symbol recognition in production of electronic navigation charts from paper charts," *J. Navig.* **77**(1), 116–123 (2024). doi: 10.1017/s0373463324000201.

[5] I. Kishor, U. Mamodiya, S. Saini and B. Bossoufi, "Voice-enabled human–robot interaction: Adaptive self-learning systems for enhanced collaboration," *Robotica* **43**, 2143–2171 (2025). doi: 10.1017/s0263574725000438.

[6] R. Ali, S. Dogru, L. Marques and M. Chiaberge, "Adaptive robot navigation using randomized goal selection with twin delayed deep deterministic policy gradient," *Robotics* **14**(4), 1–19 (2025). doi: 10.3390/robotics14040043.

[7] P. Pikulin, V. Lishunov and K. Kułakowski, "Optimizing path planning for automated guided vehicles in constrained warehouse environments: Addressing the challenges of non-rotary platforms and irregular layouts," *Robotics* **14**(4), 1–19 (2025). doi: 10.3390/robotics14040039.

[8] B. Anbarasu, "Position and heading estimation for indoor navigation of a micro aerial vehicle using vanishing point," *J. Navig.* **77**(2), 219–240 (2024). doi: 10.1017/s0373463324000286.

[9] I. Fujino and C. Claramunt, "Navigation pattern extraction from AIS trajectory big data via topic model," *J. Navig.* **76**(4–5), 506–524 (2023). doi: 10.1017/s0373463323000206.

[10] Z. Hu, Z. Yang, X. Liu and W. Zhang, "Radar-based path planning of autonomous surface vehicle with static and dynamic obstacles in a Frenet frame," *J. Navig.* **76**(4–5), 487–505 (2023). doi: 10.1017/s037346323000019x.

[11] V. Bolbot, G. Theotokatos and L. A. Wennersberg, "A method to identify and rank objects and hazardous interactions affecting autonomous ships navigation," *J. Navigation* **75**(3), 572–593 (2022). doi: 10.1017/s0373463322000121.

[12] T. Ma, W. Zhang, Y. Li, Y. Zhao, Q. Zhang, X. Mei and J. Fan, "Communication-constrained cooperative bathymetric simultaneous localisation and mapping with efficient bathymetric data transmission method," *J. Navig.* **75**(4), 1000–1016 (2022). doi: 10.1017/s0373463321000904.

[13] P. Ma, J. Yang, H. Li, Z. Zhang and H. Baoyin, "Autonomous navigation for Mars probes using only satellite-to-satellite tracking measurements by singularity-avoiding orbit elements," *J. Navig.* **75**(2), 476–495 (2022). doi: 10.1017/s0373463322000029.

[14] R. Ahmed, R. A. Osman, M. Amer and A. Ala, "Navigating urban congestion: A comprehensive strategy based on an efficient smart IoT wireless communication for PV powered smart traffic management system," *PLoS ONE* **19**(10), e0310002 (2024). doi: 10.1371/journal.pone.0310002.

[15] A. A. Taiwo, C. C. Nzeanorue, S. A. Olanrewaju, Q. O. Ajiboye, A. A. Idowu, S. Hakeem, C. G. Nzeanorue, J. C. Agba, F. P. Dayo, E. C. Enabulele, V. I. Stephen, A. Oyesanya, M. I. Ogbe, R. A. Olusola and O. B. Akinnagbe, "Intelligent transportation system leveraging internet of things (IoT) technology for optimized traffic flow and smart urban mobility management," *World J. Adv. Res. Rev.* **22**(3), 1509–1517 (2024). doi: 10.30574/wjarr.2024.22.3.1886.

[16] Z. Xia, S. Xue, J. Wu, Y. Chen, J. Chen and L. Wu, "Deep reinforcement learning for smart city communication networks," *IEEE Trans. Industr. Inform.* **17**(6), 4188–4196 (2021). doi: 10.1109/TII.2020.3006199.

[17] A. A. Musa, S. I. Malami, F. Alanazi, W. Ounaies, M. Alshammari and S. I. Haruna, "Sustainable traffic management for smart cities using internet-of-things-oriented intelligent transportation systems (ITS): Challenges and recommendation," *Sustainability-BASEL* **15**(13), 9859 (2023). doi: 10.3390/su15139859.

[18] A. K. Tyagi and N. Sreenath, "Intelligent transportation system in internet of things-based computing environment," Disruptive technologies and digital transformations for society 5.0. 265–281 (2022). 10.1007/978-981-19-7622-3_12.

[19] A. G. Samah, S. I. Ahmed and A. A. Hesham, "Intelligent traffic management system based on IoT challenges and applications," *Nile Journal for Commercial, Legal and Information Systems Science* **1**(1), 32–38 (2021). doi: 10.21608/MNSLI.2021.195172.

[20] M. A. Mondal and Z. Rehena, "An IoT-Based Congestion Control Framework for Intelligent Traffic Management System," **In:** *Advances in Artificial Intelligence and Data Engineering. AIDE 2019*, Vol. 1133, (Springer, 2021) pp. 978–981. 10.1007/978-981-15-3514-7_96.

[21] H. Dui, S. Zhang, M. Liu, X. Dong and G. Bai, "IoT-enabled real-time traffic monitoring and control management for intelligent transportation systems," *IEEE Internet Things J.* **11**(9), 15842–15854 (2024). doi: 10.1109/JIOT.2024.3351908.

[22] I. Moumen, J. Abouchabaka and N. Rafalia, "Enhancing urban mobility: Integration of IoT road traffic data and artificial intelligence in smart city environment," *Indone. J. Electrical Eng. Comput. Sci.* **32**(2), 985–993 (2023). doi: 10.11591/ijeecs.v32.i2.pp985-993.

[23] Y. Kayataş and S. Kabadayı, "An AI-Assisted Autonomous IoRT Agent for Smart Spaces," **In:** *2024 9th International Conference on Computer Science and Engineering (UBMK)*, Antalya, Turkiye (2024) pp. 613–618. doi: 10.1109/UBMK63289.2024.10773582

[24] B. Singh, C. Kaunert, S. Lal, M. K. Arora and K. Jermsittiparsert, "Intelligent Mobility Assimilating IoT in Autonomous Vehicles," **In:** *Advances in Business Information Systems and Analytics Book Series*, (2024) pp. 279–300, doi: 10.4018/979-8-3693-5498-8.ch010.

[25] Selvanathan N., "IoT-Enhanced Path Planning and Obstacle Avoidance for Autonomous Mobile Robots," **In:** *2024 3rd International Conference for Advancement in Technology (ICONAT)*, GOA, India (2024) pp. 1–6. doi: 10.1109/ICONAT61936.2024.10774737Please

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press

[26] K. Amano, A. Komori, S. Nakazawa and Y. Kato. "Adaptive Navigation Method for Mobile Robots in Various Environments using Multiple Control Policies," **In:** *IEEE 21st International Conference on Industrial Informatics (INDIN)*, Lemgo, Germany (2023) pp. 1–6. doi: 10.1109/INDIN51400.2023.10217896
[27] H. Sandhaus, W. G. Ju and Q. Yang, "Towards prototyping driverless vehicle behaviors, city design, and policies simultaneously," arXiv.Org, abs/2304.06639 (2023). doi: 10.48550/arXiv.2304.06639
[28] S. Gopal, P. Gupta, M. Sharma, D. Kaushal, S. Joshi and B. Sharma, "IOT Enabled E-Vehicles for Developing Smart Transportation System," **In:** *International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, Faridabad, India (2023) 1063–1068. doi: 10.1109/ICAICCIT60255.2023.10466202
[29] S. Kaur and S. Sharma, "Internet of Things Enabled Framework for Sustainable Mobility and Clean Environment in Smart Cities," **In:** *International Conference on Recent Trends in Artificial Intelligence and IoT*, Springer Nature, Switzerland (2023) 285–298. doi: 10.1007/978-3-031-37303-9_21
[30] D. Bose, K. Mohan, M. Cs, M. Yadav and D. K. Saini, "Review of autonomous campus and tour guiding robots with navigation techniques," *Aust. J. Mech. Eng.* **21**(5), 1580–1590 (2022). doi: 10.1080/14484846.2021.2023266.
[31] J. Vaščák, L. Pomšár, P. Papcun, E. Kajáti and I. Zolotová, "Means of IoT and fuzzy cognitive maps in reactive navigation of ubiquitous robots," *Electronics* **10**(7), 809 (2021). doi: 10.3390/electronics10070809.
[32] A. K. Jabakumar, "Edge-enabled smart traffic management system: An IoT implementation for urban mobility," *Res. J. Comput. Syst. Eng.* **4**(2), 160–173 (2023). doi: 10.52710/rjcse.85.
[33] I. Picallo, H. Klaina, P. López-Iturri, J. J. Astráin, M. Celaya-Echarri, J. J. Astráin, J. Villadangos and F. Falcone, "Development of a Cognitive IoT-Enabled Smart Campus," **In:** *14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Rome, Italy (2024) pp. 644–647. doi: 10.1109/CSNDSP60683.2024.10636428
[34] V. Kansal, A. H. Shnain, A. Deepak, A. Rana, K. K. Dixit and K. V. Rajkumar, "Deep Reinforcement Learning for IoT-Based Smart Traffic Management Systems," **In:** *7th International Conference on Contemporary Computing and Informatics (IC3I)*, Greater Noida, India (2024) pp. 1707–1713. doi: 10.1109/IC3I61595.2024.10829291
[35] D. A. Janeera, S. S. R. Gnanamalar, K. C. Ramya and A. G. A. Kumar, "Internet of Things and Artificial Intelligence-Enabled Secure Autonomous Vehicles for Smart Cities," **In:** *EAI/Springer Innovations in Communication and Computing*, (2021) pp. 201–218. doi: 10.1007/978-3-030-59897-6_11.
[36] S.-H. Joo, S. Manzoor, Y. G. Rocha, S.-H. Bae, K.-H. Lee, T.-Y. Kuc and M. Kim, "Autonomous navigation framework for intelligent robots based on a semantic environment modeling," *Appl. Sci.* **10**(9), 3219 (2020). doi: 10.3390/app10093219.
[37] M. Mohamed and K. Alosman, "An IoT-enabled framework for smart city infrastructure management," TechRxiv, (2024). doi: 10.36227/techrxiv.170906134.46169506/v1.
[38] R. Parvathi, T. Ch Anil Kumar, A. Nandhakumar, M. H. Arshad and B, "Ajith Singh,"Intelligent tracking and navigating moving objects in a smart environment using IoT Network," *ECS Trans.* **107**(1), 5687–5691 (2022). doi: 10.1149/10701.5687ecst.
[39] E. Lee, Y.-D. Seo and Y.-G. Kim, "Self-adaptive framework with Master–Slave architecture for internet of things," *IEEE Internet Things J.* **9**(17), 16472–16493 (2022). doi: 10.1109/jiot.2022.3150598.
[40] J. Nagalapuram and S. Samundeeswari, "A framework for smart city traffic management utilizing BDA and IoT," *Eng. Technol. Appl. Sci. Res.* **14**(6), 18989–18993 (2024). doi: 10.48084/etasr.8003.
[41] S. S. Anjum, I. Javid, K. H. Keoy, P. Kumar, A. S. Shibghatullah, E. Natarajan, R. M. Noor and I. Ahmedy, "*An Internet of Things-based Integrative Safety Framework of Autonomous Vehicles for Special Needs Society* (Elsevier eBooks, 2024) 137–148). doi: 10.1016/b978-0-443-18644-8.00006-x.
[42] M. Hosseinzadeh, A. Haider, A. M. Rahmani, F. S. Gharehchopogh, S. Rajabi, P. Khoshvaght, T. Porntaveetus and S.-W. Lee, "SDN-based NFV deployment for multi-objective resource allocation in edge computing: A deep reinforcement learning for IoT workload scheduling," *Sustain. Comput.: Inform. Syst.* **45**(101218), 1–18 (2025). doi: 10.1016/j.suscom.2025.101218.
[43] A. M. Rahmani, A. H. Malik, K. Moghaddasi, F. S. Gharehchopogh, K. Aurangzeb, Z. Liu and M. Hosseinzadeh, "Self-learning adaptive power management scheme for energy-efficient IoT-MEC systems using soft actor-critic algorithm," *Internet of Things* **31**(101587), 1–21 (2025). doi: 10.1016/j.iot.2025.101587.
[44] S. R. Moghaddasi, F. S. Gharehchopogh and A. Ghaffari, "An advanced deep reinforcement learning algorithm for three-layer D2D-edge-cloud computing architecture for efficient task offloading in the internet of things," *Sustain. Comput.: Inform. Syst.* **44**, 100992 (2024). doi: 10.1016/j.suscom.2024.100992.
[45] K. Moghaddasi, S. Rajabi and F. S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled IoV-MEC systems: A double deep Q-network approach," *IEEE Access* **12**, 3437–3463 (2024). doi: 10.1109/ACCESS.2023.3348513.

Cite this article: I. Kishor, U. Mamodiya, S. Saini, M. A. Almaiah and V. Gandhi (2025). "An Internet of Things-enabled adaptive robotic framework for autonomous navigation in smart campus and urban mobility systems", Robotica **43**, 4268–4301. https://doi.org/10.1017/S026357472510283X

https://doi.org/10.1017/S026357472510283X Published online by Cambridge University Press