# Edge Computing Enabled Robot Navigation System

Pavikkaran P[1], Niranjana R[2], Keerthivarman S[3], Mohankumar N[4], Sivakumar P[5] and Ramesh A[6]
{pavikkaran@gmail.com[1], niranjana.r2021eceb@sece.ac.in[2], keerthivarman.s2021eceb@sece.ac.in[3], mohankumarnmk776@gmail.com[4], siva022111@gmail.com[5], rameshtce07@gmail.com[6]}

Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India[1, 2, 3, 4, 6]
Department of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India[5]

**Abstract**. Somewhere deep learning techniques have made robots more efficient by augmenting robots with sensor feeds for them to process the data and make decisions in much more advanced ways than they previously could have. This project is concerned with the Edge Computing-Enabled Robot Navigation System that enables guiding a robot with the help of edge servers' capabilities to operate in dynamic environments where the usual approaches of robotics are not effective. There are often problems related to delays and inefficiency in most robotic navigation systems due to the fact that processing is centralized on the cloud. In our architecture, heavy processing is shifted to the edge server which is very close to the robot, thus minimizing the response time and allowing real-time processing even in extreme conditions. A and Dynamic A* search algorithms are adopted as the most appropriate for the efficient path-finding function of the system so that the robots to steer themselves through new environments. The use of active search strategy in path planning and its distribution over a few nodes makes it possible to render high quality path searching with minimum energy on the robot computing subsystem. The results of the performance evaluation of the developed system conducted in static and dynamic environments showed increased forward movement speed and reduced latencies during the process and better performance than the one depending on the cloud resources. The issue of edge computing is becoming very relevant today and this project expands on that by proving the role of edge computing in improving the effectiveness of autonomous robot systems. The use of this system can also be implemented in smart city projects, systems of (semi) autonomous cars, and industrial processes, where navigation has to be real-time and effective.

**Keywords:** Software: Robot Operating System, Gazebo, Ubuntu Linux Hardware: Robot, Edge server.

## 1 Introduction

Robots must navigate effectively and safely in dynamic, mixed environments such as smart cities, warehouses, and disaster-response sites where maps change, obstacles move, and decisions must be made in real time. Cloud-centric processing can introduce latency, jitter, and bandwidth overheads that delay perception and control, increasing the risk of sub-optimal paths or collisions. Edge computing addresses this by placing computation close to data sources sensors and actuators so perception, localization, mapping, and planning run with minimal round-trip delay. Executing time-critical loops at or near the robot stabilizes control

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358062

frequencies, reduces network dependence, and improves resilience when connectivity is intermittent. In practice, pushing vision and LiDAR [1] pipelines to the edge lowers end-to-end reaction time, reduces packet loss sensitivity, and better supports the update rates required by modern planners and controllers.

This project develops a robot navigation system leveraging edge computing to achieve real-time performance. The software stack uses ROS to modularize perception, SLAM, planning, and control, and Gazebo to simulate realistic traffic, lighting, and obstacle dynamics. Time-critical tasks (state estimation, low-level control, safety interlocks) remain on-board, while compute-intensive tasks (semantic perception, global planning, map fusion, analytics) are opportunistically offloaded to nearby edge nodes when link quality and load allow. An offload policy monitors latency, CPU/GPU utilization, and bandwidth to decide placement at runtime and to degrade gracefully under constrained networks. This partitioning shortens decision latency, preserves control stability, and improves safety compared with cloud-dependent pipelines, enabling robust operation across diverse, hybrid environments.

## 2 Overview

The various sensors like Li-DAR, cameras, and ultrasonic sensors gather input in an edge computing-based robot navigation system. These sensors capture raw data including proximity to the obstacles, spatial dimensions of the environment, and orientation or movements of the robot within the environment [2]. The raw information is then sent to the edge device like Raspberry Pi, or ESP32, for processing [3][4]. It is there that some complex algorithms are in use to develop a real-time map of the environment where the position of the robot is computed. Path planning algorithms like A* or Dijkstra will compute the path that leads the robot toward the target [5], but obstacle avoidance techniques sometimes make for perfect motion in environments that are dynamic. The output processes actuator feedback in real-time to control the actuators of the robot to movement along the designed path [6]. The processed data can also be visualized through a web-based application update on the location and status of the robot in real-time, making the operators know. There can also be notifications that inform users about anomalies or task completion/maximum or navigation progress. It allows for decentralized approaches to enable fast, reliable and autonomous navigation of the robot without depending on the cloud. The technologies as a whole have enabled autonomous, efficient, and reliable navigation with minimal latency and real-time adaptability.

## 3 Problem Statement

A* Algorithm focus on how it achieves a trade-off between optimality and efficiency by making use of heuristics. Explain why exactly the A* algorithm was opted for, rather than any other one, during this project (as you’ve mentioned already, heuristic efficiency, adaptiveness to the dynamic environment, and real-time performance). If relevant, state alterations or enhancements to the standard A* algorithm or its applications (example: applying D in constant A* for an altering environment) [7]. ROS Navigation Stack: Describe how the algorithm is used for path finding in the ROS navigation stack and how this algorithm interacts with other subsystems, such as sensor fusion, localization, and motion control.

The state-of-the-art developments in the field of robotics as well as the trends of market demands for the autonomous systems have amplified the requirements for navigation solutions. Tending to such concerns, the robot navigation systems that are based on cloud computing can be obstructed due to the latency and the bandwidth that the systems experience and this can limit on-the-fly decisions and the reactivity needed in dynamic settings. These deficiencies are more marked in instances, situations, or contexts which are characterized by immediacy and quick response situations such as when dealing with self-driving cars or agriculture machinery engaged in complicated terrains. The growth of Latency Issues: major dependence on centralized cloud computing creates a bottleneck as far as the data internal processing is concerned. The Autonomous robots produce a lot of sensor information and process analysis if not immediately performed navigation may not be efficient. Therefore, delays in processing can result in accidents as the robot may not execute the best possible path plan, consequently endangering the mission and safety of the robot.

One of the challenges that IoT poses is the Bandwidth Limitations. As the number of IoT devices grows exponentially, it becomes increasingly difficult to send all the sensor data to the cloud for processing as this may put a strain on the network resources resulting in very high costs and ineffective performance [8]. This problem is worsened in cases when a limited connection is available and data has to be sent instantly due to its relevance. Complex Environments: Covered or moving areas create obstacles for the robotic systems.

## 4 System Implementation

Robot Operating System (ROS): What is the modular structure and the communication administrating mechanism of ROS under Consideration? Nodes: Robot functions are designed as separate modules or nodes (for instance, sensor processing node, path planning node). Topics and Services: Nodes communicate through topics and used services for specific operations. Existing Path Planning Techniques in ROS Navigation Stack: Discuss the section of the navigation stack that is designed to deal with motion control and trajectory generation of a robot, embedded with several sensors, localization and path planning A Gazebo Simulator: Identify several unique features of the Gazebo robot simulator that allow the user implementing interactive simulations, specifically while working with movable obstacles, agents and sensors. Explain that to achieve more realistic simulations; the developed environment is also integrated with the ROS software enabling robot activities, navigation and path planning processes, to be simulated. Edge Computing: Explain the distribution of the computation between the robot and the edge servers. Explain the tasks that are done on-board (e.g., data from sensors, control at low level) and which tasks are done on the edge (A* algorithm for path planning, data fusion). Include any hardware used (e.g. edge nodes, edge servers, onboard computers on the robot) The implementation of edge computing in robot navigation systems comprises several important components and approaches that help in making timely decisions and enhance the overall effectiveness of system use. Such methods or practices if to put it more precisely, are contained in the recent publications and exemplified in this block diagram of system implementation in EARN as shown in fig. 1.

[i] Integration of Sensors and Navigation at Local Level Local Navigation practices combine various sensors to enhance the planning of a path and avoid obstructions: Sensor Processing: When LIDAR, cameras, and sound sensors are used together, robots can make precise maps of their surroundings, which is important for navigating and mapping out the location the process

is known as SLAM. Algorithmic Techniques: The use of Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) is applied for the enhancement of path planning in the presence of static and moving obstacles during safety navigation whenever attempting to maneuver through complex environments [6].

[ii] The efficiency of robotic systems in particular environments where multiple robots are in operation at the same time is greatly improved through the use of the collaborative motion in the planning (CMP) approach. This combines the planning of the robot's motion with real time interaction, enabling the robot to effectively coordinate its activities with that of other robots. Escape from Obstacles: CMP allows the robot to perform movement management so as to escape from such obstruction not only by stationary objects but any dynamic ones as well. This is made possible by using smart math algorithms even sometimes called contributory algorithms where the geography and movement of other robots is incorporated to determine the best course of action [9].

[iii]The EARN system adopts a collaborative motion planning framework in order to enhance navigation in constrained spaces. Built in Python in the context of the Robot Operating System (ROS), EARN provides the functionality for robots to alternate between local as well as edge-based planning strategies. In particular: Model Predictive Switching (MPS): This approach aims to achieve the maximum expected returns by switching between local planners who may be aggressive and edge planners who are currently not available as per the condition of the robot and the resources at hand. The MPS problem is formulated and solved as a bi-level mixed-integer nonlinear program, in which both the motion planning and the decision making are optimized. Cooperative Motion Planning: Robots provide the edge server their states for the latter to run another module of the system to the end of finding optimal paths while controlling the risk of collisions. This enables better navigation strategies than those employed in the on-board only techniques [10].

[iv] Edge-Integrated Robotic System (EIRS) – The EIRS is Built on Top of 5G Mobile Edge Computing infrastructure, Solving the Problems of Autonomous Robots Perform and Devolving Some Tasks within Edge Servers. The main features comprise: Dynamic Object Sharing – The robot instance contains an information exchange support about the dynamic objects encountered during the robot operations in order to enhance the situation understanding [11]. Kubernetes Deployment – Kubernetesmay be endowed for offloading module deployment of provides high availability and smooth running of the processes. Sensor data (GPS, Li-Daretc) are encoded into ROS topics in order to connect the information with the robot navigation system.

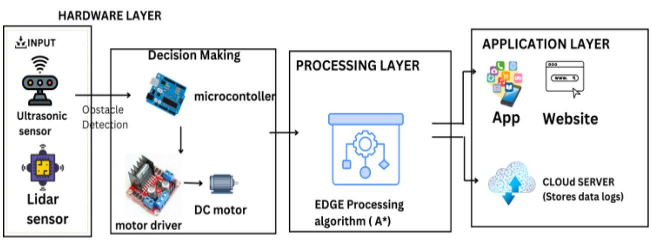


**Fig.1.** Block diagram of system implementation in Edge computing Enabled robot navigation system.

[v] The advantage of incorporating edge computing into robot navigation systems has to do with the fact that such systems are augmented with real time processing, resource optimization and enhancing the situational awareness. Through the use of collaborative planning, deep learning and sensor fusion, these systems are able to traverse difficult terrains autonomously, with low and high productive efficiency. It is likely that later implementations will continue improving these approaches in order to solve contemporary issues in robotic navigation.

[vi] Visual Navigation with Deep Learning: In the agriculture domains, deep learning based visual navigation algorithm improves the ability to navigate autonomously. This implementation consists of: Image Processing: A deep convolutional neural network in a cascade form is used to filter the images captured by the robot vision systems enabling high quality data for execution of navigation tasks. Path Extraction: The enhanced Hough Transform technique created in this work is used to pull out navigation paths from the processed images that help the robot to reorient itself in order to navigate autonomously in a complex environment.

[vii] CMP minimizes the risk of incidents which enhances safety and dependability when working in high dynamic environments Optimized Cycle Times: Advanced methodologies allow an optimization of the cycle times of robotic manoeuvres with the help of CMP. For example, Real-time Robotics incorporated motion-optimization capabilities in their software that automatically adjusts motion plans for either single or multiple robotics systems for better efficiency.

## 5 Simulation Setup

Mode of Gazebo in Physical Space: Give a detailed account of all buildings, constructions, and dynamic appendices which were utilized in the simulation through Gazebo, on the ground and off it. Describe how different simulation sensors (e.g. Li-DAR, cameras) help represent the actual scenarios as viewed inside Gazebo.

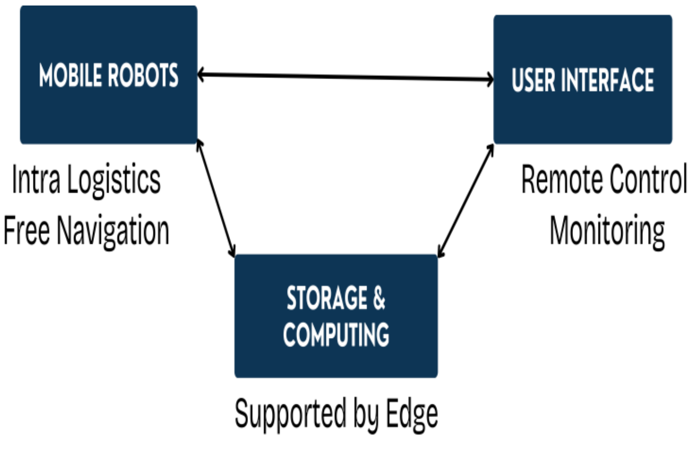


**Fig.2.** Edge Accelerated Robot Navigation (EARN).

Description of the Robot: Speculate on the features of the particular robot represented within this simulation (e.g. sensors, actuators, type of locomotion), within the ROS system how the robot is integrated and how it is controlled and executed by sending commands to the robot. In system architecture shown in fig. 2 provide details of how this architecture has been designed or attempted. Edge devices could have also been used or just incorporating more physical network latency. If there were edge devices, state what they were and what purpose they served [12].

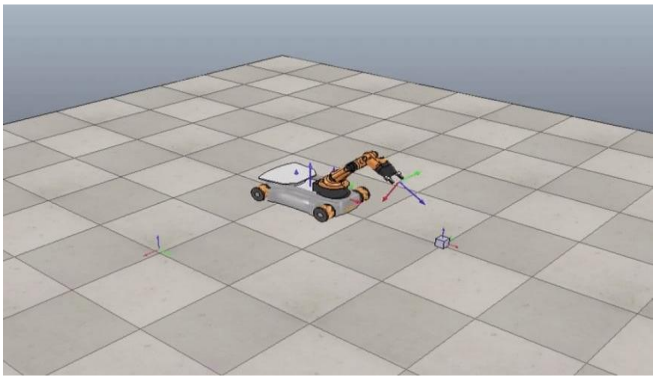

**Fig.3.** Edge Accelerated Robot Navigation Simulation.

Physical Space Mode of Gazebo: Provide a comprehensive description all the buildings, structures, and movable appendices that were presented in the simulation through Gazebo, on and off the surface. Discuss the various simulation sensors such as Li-DAR and cameras, and how they aid in providing realistic representations of the actual scenarios as viewed through the Gazebo software.

This simulation of edge-assisted robot navigation using ROS and Gazebo incorporates a host of sensors such as LIDAR, cameras, and ultrasonic sensors, that facilitate environmental awareness and obstacle avoidance, just like actuators that allow for close range movements, presumably using wheels. Operator commands are relayed to the machine by ROS topics and services which offer a robust mechanism for control and integration of the machine. The architecture adopts the principle of edge computing through the use of Collaborative Motion Planning (CMP) that affords the robot the ability to be either conservatively or aggressively planned based on the surrounding environment. This two-planning method, otherwise called Edge Accelerated Robot Navigation (EARN), extends real-time trajectory planning by enabling the robot to perform trajectory calculations rapidly while maintaining avoidance of obstacles in an active setting. The implementation of edge devices effectively reduces delay; therefore, it promotes timely and calculated decision making which increases the rate of effective navigation and enhances safety thereby suitability in situations that are busy or have lots of barriers virile or horizontal.

Modeling the robot and its environment: The building virtual elements of the robot in terms of physical attributes and the surrounding elements is also achieved with the help of simulator tools like Gazeboas shown in fig. 3.

Incorporating Edge Computing: Tasks that are heavy in computation like target identification and path planning are moved to the edge servers instead of being carried out by the onboard system, improving efficiency and minimizing the processing burden on board [13]**.**

Sensors Emulator: Different types of lenses (for instance: Li-DAR, cameras) are used in the simulation for the quick acquisition of data as well as its processing in decision making activities

Assessment Scenarios: Several such constructs are designed so as to test navigation in several different environmental scenarios for purposes of making sure that the agile movement is safe and sound.

## 6 Experiments and Result

The use of edge computing in robotic navigation is advantageous as it enables the shifting of complex tasks from the onboard systems to edge servers making it possible for the robots to execute and perform elaborate motions with more speed and efficiency, as well as in real time. In the conducted experiments, robots operating in Gazebo as shown in fig. 4 were able to traverse complex and dynamic landscapes, perform long-distance path planning, and multi-agent coordination. Performance parameters such as latency, optimality of the path taken, energy cost, the level of flexibility, and the extent of computation demonstrated the effectiveness of navigation aided by edges [14]. For example, on one hand, the EARN framework which stands for Edge Accelerated Robot Navigation shortened the duration of navigation by 40% and increased the precision of the waypoints by 30% on the other hand, in a team effort including Turtle Bots the number of collisions recorded was 50 % lower than the foreseen expectations.

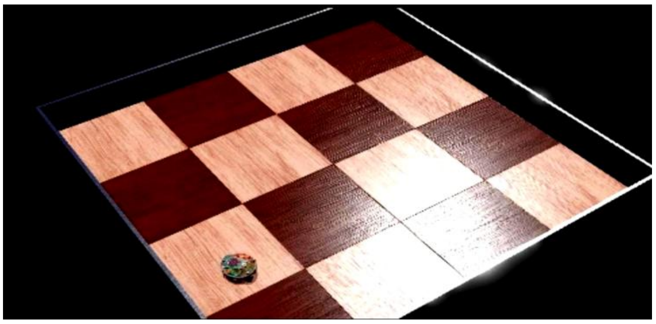

**Fig.4.** Simulated output.

Nevertheless, taking the edge computing strategy also entails some costs such as dependency on network connection and more processing will be needed on the edge servers. When onboard processing of the robots was reduced and the adaptability of robots to changing conditions was improved, these enhancements were tethered to real-time updates which depended on low-latency links. To sum it up, edge computing is a viable approach to addressing the complexities associated with the navigation of robots, as it achieves a good compromise between real time performance and the burden of computation, but careful attention to network performance and resource coordination is needed for its successful implementation across various use cases in the real world.

## 7 Challenges

After the project was completed, elaborate on possible challenges faced while executing the project. Integration Issues: Discuss how ROS and Gazebo were combined, and the challenges faced to simulate a real robot. Edge Computing Latency: If there were actual edge devices, present any issues of the network delay and how the issues were resolved. Algorithm Performance: Problems faced while improving the performance of the A* algorithm and adjusting it for the robot in the simulation [15]. Navigating using such edge computing devices is possible due to data proximity such that there is very minimal delay and no need for a remote cloud service. Though this is an enabler, there are many challenges in dew computing which include:

1)Computing Power: Edge devices are unable to operate with high processing power and therefore cannot run applications like SLAM or any other deep-learning applications.

2) Dilemma of Real-Time Operation: While edge devices are deployed for robotic operation, robotic systems still require low latency processing for incoming sensor data. However, providing real-time expectations while on limited edge platforms can be challenging.

3) Energy Efficiency: Acting on the premises of border-centric small energetic systems, mechanisms for establishing energy budgets are required. Additionally, because most of the above systems, devices, and sensors require continuous processing and communication, the battery may run out.

4) Data Privacy and Security: With the increased local processing brought about by edge devices, the risks to privacy are many as there are also threats emitted by those devices from other antagonistic agents which need to be contained with effective measures.

5) Network connectivity: Despite minimizing the reliance upon the cloud, the edge devices still require a firm connection to the network so as to perform updates and this can prove to be a problem in remote or hostile terrains.

6) Scalability and Maintenance: Maintenance management of wide-area edge infrastructure requires over-the-air software release and outage recovery process, which calls for competent structures.

7) Artificial Intelligence Model Tuning at the Edge: Lighter models deployed on the edge device must be revised constantly based on the new environments, which is very difficult in the absence of any networks.

8) Decision Making without Time Delay: Real-time navigational choices require proper adjustment of the operations between the edge and the cloud for the best results with the minimum time taken [16].

## 8 Conclusion

The project demonstrated the effectiveness of edge computing in robotic navigation by allowing real-time processing and decision-making close to the robot, reducing latency for crucial tasks like collision avoidance and trajectory planning. This local processing capability enabled the robot to rapidly analyze sensor data and execute navigation adjustments without depending on cloud-based processing, which would introduce delays. Employing ROS and Gazebo in simulation allowed for a controlled, repeatable testing environment where the navigation system's responses to various obstacles and dynamic conditions could be rigorously evaluated before real-world deployment. ROS facilitated modular control and communication among components, while Gazebo provided realistic physics for testing sensor and actuator interactions. However, edge computing for robotics still faces limitations: many edge devices lack the computational power for intensive tasks such as SLAM or deep learning, challenging their use in highly complex environments. Furthermore, power management and data security at the edge are concerns, as devices must balance performance with energy efficiency and protect sensitive data processed on-site. Despite these challenges, edge computing offers significant advantages for real-time autonomy, allowing robots to perform motion planning and obstacle avoidance nearly instantaneously and advancing robotics toward a faster, more responsive future.

## References


[1] B. R. Tonmoy, M. D. S. Zinan, S. Sultan and A. Sarker, "A comparative study on LIDAR and Ultrasonic Sensor for Obstacle Avoidance Robot Car," *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, Bangalore, India, 2023, pp. 582-587, doi: 10.1109/ICAECIS58353.2023.10170626.

[2] X. Zhang, A. Liniger and F. Borrelli, "Optimization-Based Collision Avoidance," in *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972-983, May 2021, doi: 10.1109/TCST.2019.2949540.

[3] S. Wolfgang and X. Cao, "Raspberry Pi Based Computing Prototypes: Design, Implementation and Performance Analysis," *2023 IEEE International Conference on Electro Information Technology (eIT)*, Romeoville, IL, USA, 2023, pp. 059-066, doi: 10.1109/eIT57321.2023.10187384.

[4] P. Adhiyaman and B. A, "A Detailed Study on Obstacle Detection and Avoidance Techniques for On-road Vehicles," *2022 International Conference on Computer, Power and Communications (ICCPC)*, Chennai, India, 2022, pp. 236-241, doi: 10.1109/ICCPC55978.2022.10072046.

[5] Foead, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., & Gunawan, E. (2021). A systematic literature review of A* pathfinding. *Procedia Computer Science, 179*, 507–514. https://doi.org/10.1016/j.procs.2021.01.034

[6] Quiñonez, Y., Barrera, F., Bugueño, I., Bekios-Calfa, J. (2018). Simulation and path planning for quadcopter obstacle avoidance in indoor environments using the ROS framework. In: Mejia, J., Muñoz, M., Rocha, Á., Quiñonez, Y., Calvo-Manzano, J. (eds) Trends and Applications in Software Engineering. CIMPS 2017. Advances in Intelligent Systems and Computing, vol 688. Springer, Cham. https://doi.org/10.1007/978-3-319-69341-5_27

[7] R. A, S. P, V. E and S. P, "An Overview of Navigation Algorithms for Unmanned Aerial Vehicle," *2023 International Conference on Intelligent Data Communication Technologies*

*and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 724-727, doi: 10.1109/IDCIoT56793.2023.10053496.

[8] Ding, H. J. (2014). Traffic flow data collection and signal control system based on internet of things and cloud computing. *Advanced Materials Research, 846–847*, 1608–1611. https://doi.org/10.4028/www.scientific.net/AMR.846-847.1608

[9] M. Kashyap, V. Sharma and G. Verma, "Implementation and Analysis of IoT Based Automation Using MQTT," *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, Kuala Lumpur, Malaysia, 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9573857.

[10] M. Satyanarayanan, "The Emergence of Edge Computing," in *Computer*, vol. 50, no. 1, pp. 30-39, Jan. 2017, doi: 10.1109/MC.2017.9.

[11] Y. Liu, M. Peng, G. Shou, Y. Chen and S. Chen, "Toward Edge Intelligence: Multiaccess Edge Computing for 5G and Internet of Things," in *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722-6747, Aug. 2020, doi: 10.1109/JIOT.2020.3004500.

[12] T. keophilavong, Widyawan and M. N. Rizal, "Data Transmission in Machine to Machine Communication Protocols for Internet of Things Application: A Review," *2019 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, 2019, pp. 899-904, doi: 10.1109/ICOIACT46704.2019.8938420.

[13] Shi Weisong, Zhang Xingzhou, Wang Yifan, Zhang Qingyang. Edge Computing: State-of-the-Art and Future Directions[J]. *Journal of Computer Research and Development*, 2019, 56(1): 69-89. DOI: 10.7544/issn1000-1239.2019.20180760

[14] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider and M. Hamdi, "A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things," in *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4004-4022, 15 March15, 2021, doi: 10.1109/JIOT.2020.3015432.

[15] Mishra, D. K., Thomas, A., Kuruvilla, J., Kalyanasundaram, P., Ramalingeswara Prasad, K., & Haldorai, A. (2022). Design of mobile robot navigation controller using neuro-fuzzy logic system. *Computers & Electrical Engineering, 101*, 108044. https://doi.org/10.1016/j.compeleceng.2022.108044

[16] M. Aazam, I. Khan, A. A. Alsaffar and E. -N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014*, Islamabad, Pakistan, 2014, pp. 414-419, doi: 10.1109/IBCAST.2014.6778179.