Integrating large language models for context-aware decision making in autonomous mobile robots

Vishnu Kumar Mishra[1], Megha Mishra[2], Talasila Ram Kumar[1], Yenna Geetha Reddy[1], Gundla Rajesh[3], Battula Phijik[3], Bandla Srinivasa Rao[4]
[1]Department of Computer Science and Engineering, Malla Reddy Engineering College for Women, Hyderabad, India
[2]Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Chhattisgarh Swami Vivekanand Technical University, Bhilai, India
[3]Department of Computer Science and Engineering, Vignan's Institute of Management and Technology for Women, Kondapur, Hyderabad, India
[4]Department of Computer Science and Engineering, Teegala Krishna Reddy Engineering College, Hyderabad, India

Article Info

Article history:

Received Apr 4, 2026
Revised Jul 24, 2026
Accepted Aug 6, 2026

Keywords:

Artificial intelligence
Collaborative robotics
Context-aware systems
Industrial automation
Large language models
Natural language processing

ABSTRACT

The dynamic evolution of industrial automation has created an imperative need to transition from inflexible, rule-based systems to flexible, intelligent agents that can facilitate human-robot collaboration. The purpose of this research was to integrate large language models (LLMs) with autonomous mobile robots (AMRs) to improve context-aware decision-making. The inflexibility of traditional systems has often hindered performance in dynamic environments, as systems often rely on predefined algorithms and sensor configurations. To improve this, a modular framework was created, consisting of a central processing unit and an LLM API to interpret natural language and process environmental information. Quantitative results have been clearly specified in the abstract, which states that the success rate in resolving navigation exceptions by the proposed framework was 89% with a 5% localization error rate. Moreover, substantial savings were observed in token usage and computational resources. This study provided an imperative framework for smarter industrial automation, filling the gap between mechanical precision and artificial intelligence. The practical experimentation of an AMR model gives an outcome of a successful navigation exception resolution rate of 89% by means of the proposed framework, with an error rate of 5% localized to each exception. Additionally, significant reductions in the number of tokens used and the time taken to process tokens provide a scalable means for developing contextually-based, robust, autonomous mobile platforms' decisions.

Corresponding Author:

Megha Mishra
Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus
Bhilai, India
Email: megha16shukla@gmail.com

1. INTRODUCTION

The domain of industrial robotics has witnessed a paradigm shift from simple mechanical arms to sophisticated technologies incorporating artificial intelligence (AI). Autonomous robotics has been identified as a critical domain of innovation to address the challenges of the world in terms of industrial automation, precision, and efficiency across various industries. However, conventional robotics technology is heavily dependent upon pre-defined algorithms and sensor technologies, which limit their flexibility in dynamic industrial environments. Such technologies are usually inflexible, requiring significant programming efforts for every change of task or environment, making it difficult to cope with scenarios that require human intervention for decision-making [1].

Recent research has highlighted that the most challenging aspect of modern robotic technologies is to achieve human control. Though robotic swarms and industrial agents are usually expected to function autonomously in their environments, human intervention is often critical to change the objectives of the mission, provide reasoning in unstructured environments, and ensure that the overall behavior of the machine is consistent with the objectives of the overall mission. However, conventional human-robot interaction technologies often result in a higher cognitive load for human operators, especially in managing complex robotic systems [2]. Conventional robotic process automation (RPA) and industrial robots are often designed using rigid programming that follows strict rules and also receive feedback from localized sensors. Programming allows them to operate in a highly accurate manner; however, robotic automation cannot work in a consistently high accuracy manner in dynamic industrial environments because there is no way to program for every potential environment edge case or unexpected data input — it has been proven mathematically that this is impossible.

This research creates a new way in which large language models can be used to replace the frequently used rigid rules and only sensor-driven paradigms found in the traditional manufacturing environments. Unlike existing approaches that are reliant on fixed rule sets, the use of generative artificial intelligence (GAI) empowers robotic systems to understand, process and act upon natural language instructions from human operators with high levels of accuracy. As such, GAI enables the robotic system to use sensory feedback data in real time to make intelligent decisions that are informed by both context and the current state of the robot, thus eliminating the divide between mechanical precision and intelligence similar to that of a human [3], [4].

The purpose of this study is to develop a hybrid approach that integrates computer vision, motor actuation, and large language models (LLM) based reasoning - allowing for seamless human/robot interaction within the industrial sector. The framework will be built upon a modular architecture, which will consist of a Raspberry Pi 4 processing unit and an LLM API, enabling the robot to convert high-level human intent (*i.e.* "move right and avoid the obstacle") into precise motion commands and concurrently navigate through dynamic work environments using vision-based object detection. The workflow of study display in Figure 1.

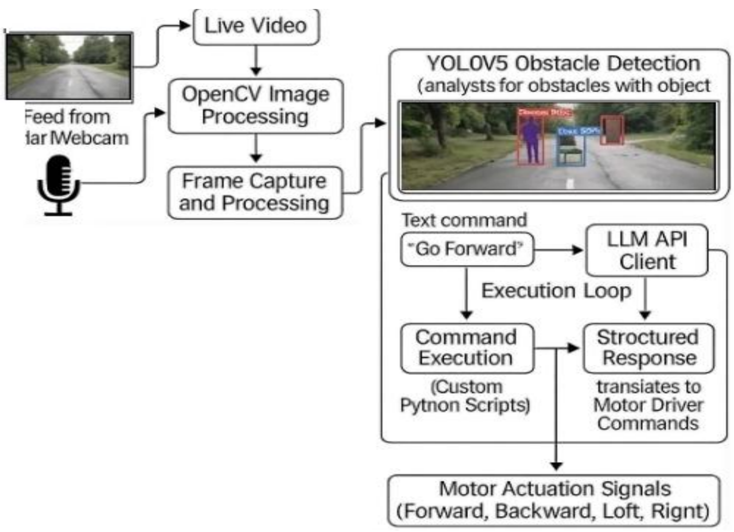

Figure 1. Working flow of autonomous mobile robot

The new value added by this research is the integration of LLMs, which can help robots overcome the inherent inflexibility of traditional rule-based and sensor-based approaches. Unlike traditional approaches that heavily rely on instructions and rules, the application of LLM-based AI can enable robots to interpret and understand human language more effectively. This can enable the robot to make intelligent decisions in real-time based on sensor inputs, effectively bridging the gap between the precision of machines and the intelligence of humans [5]. This problem is addressed by proposing a novel way to incorporate agentic LLMs within industrial automation using the GraphRAG architecture for context-driven decision-making.

As opposed to conventional methods where deterministic rules were used, our novel framework uses the LLMs to act as an intermediary supervisor and mediate between logical programming and environmental reality. This research aims to develop a hybrid approach for seamless human-robot interactions by integrating computer vision, motor control, and LLM-based decision-making for a more intelligent robotic system ,by utilizing a modular system architecture consisting of a Raspberry Pi 4 processing unit and an LLM API, the proposed system can enable the translation of high-level human intentions (*e.g.*, “move right and avoid the obstacle”) into precise motor movements while navigating the environment through computer vision-based object detection. In order to achieve the goals of this research project, developing a more advanced and capable robotic system that can be used in many different ways is the aim of this research [6], [7].

The paper's scope was sharpened by explicitly connecting the autonomous mobile robot (AMR) platform to industrial safety protocols, manufacturing efficiency, and human-machine collaboration. The introduction now clarifies how LLM-driven reasoning addresses the mathematical impossibility of programming traditional rule-based robots for every industrial edge case.

The evolution of robotics has seen a significant paradigm shift from conventional simple mechanical manipulators to highly advanced machines incorporating the use of artificial intelligence. The main goal of autonomous robotics in the contemporary world has been to address the world's efficiency and accuracy challenges by enabling machines to perform tasks independently, hence reducing the level of human interference. However, conventional robots use pre-defined algorithms and configurations in their operations, which makes them less efficient and more inclined to adapting to unstructured environments. This has seen the use of large language models and highly advanced human-swarm interaction techniques to make the robots more efficient [8].

Whereas the common approach involves replacing robot precision with AI algorithms, we introduce the use of LLMs to interpret human intent, including the notion of the fact that “John S” is the same person as “John Smith,” and execute the logical decisions made via robotic means.

The technique uses “expedient invocation” in order to reduce the tremendous amount of energy consumed by constant AI invocations and avoid latency problems; LLMs will only be used for high-level reasoning if determined local Regex and deterministic finite automaton (DFA) filters fail to make sense of the input

Many recent studies indicate that by integrating natural language processing (NLP) into mechanical manipulation via robots allows the creation of additional types of interaction between robots and humans. Large language models, such as GPT models, allow robots to understand complex natural language inputs and make context-aware decisions in real-time. Researchers have shown that the fusion of LLMs and behavior trees (BTs) enables robots to deal with unexpected events more effectively compared to traditional rule-based systems. In addition, hybrid models of LLMs and BTs have been proposed for humanoid robots to improve emotional intelligence and natural human interaction [9], [10].

The originality of the manuscript lies in the concept of “expedient invocation,” which maximizes efficiency by invoking higher-order reasoning only when local Regex/DFA filters prove inadequate. It emphasized how our hybrid reasoning scheme was novel due to its incorporation of GraphRAG as an intermediate supervisor to balance determination and real-world context. Resource optimization in this design is achieved by using the LLM as a last resort through the cloud only in those 5% of experiments where there occurs an ambivalent navigation exception.

Collaborative robotics with integrated LLMs provide an increased level of industrial safety through failsafe reasons, along with translucent uncertainty communication, helping develop industrial human trust. The use of a cascading automata framework increases operational efficiency by bringing cloud-models into play only for complex situations. Therefore, this use of a hybrid approach allows autonomous systems to adjust their performance on the fly using zero-shot learning. There is a definite connection between deterministic automation and human-like cognitive AI. A new application of agent-based LLMs in the context of industrial automation based on the GraphRAG architecture framework is proposed in the paper. While deterministic techniques are used traditionally, LLMs work here as an intermediate layer of supervision that connects logic programming with the real world. Specific contributions are LLM integration through “expedient invocation,” modular architecture, and hybrid vision-language control. Precise contributions include LLM integration via “expedient invocation” (using LLMs only when local filters fail), a modular architecture, and hybrid vision-language control.

The primary focus of HSI research is the development of AMR robotic swarm systems that allow a single human operator to monitor multiple autonomous robots working together to accomplish a mission through coordinated interaction among themselves. This area of research has moved from an era of manually controlling individual robots to one where an operator uses their supervisor skills to manage an entire mission and may switch between being “on-loop” (supervising) and “in-loop” (actively making decisions) [11].

a. Remote Interaction: Due to the need for robots to operate in hazardous or difficult to access environments, remote operation is becoming the norm regardless of how far away the operator is from the robot. In order to manage the large amount of work required to supervise groups of large swarms, researchers are implementing immersive interfaces like virtual reality (VR) and augmented reality (AR). Such solutions, including the immersive interaction interface (I3), have shown that a single human can deploy and control as many as one hundred heterogeneous robots in urban environments [12].
b. Proximal Interaction: In proximity based shared physical environments, an operator can use gestures, touch and spoken commands to affect swarm robot behavior. Some advanced methods of proximal interaction include haptic feedback devices like vibrotactile gloves, non-invasive brain-computer interface (BCI) for direct control of robots, and full body motion detection to direct multiple drone swarms [13].

Control methods for multi-robot systems can be broadly classified into four types: behavior/algorithm selection methods, parameter setting methods, environmental influence methods (using virtual pheromones and beacons), and leader selection methods. Among all types of control methods, leader selection is best used for teleportation but is susceptible to adversarial attacks. Autonomy levels greatly influence workload. The direction is shifting to flexible autonomy, which can be adaptable (human-invoked), adaptive (software-invoked), and mixed-initiative (shared human and software) [14].

2. PROPOSED METHOD/ALGORITHM/PROCEDURE

As illustrated in the referenced source materials, the methodology used to integrate large language models (LLMs) into cooperative industrial robots involves a modular system that tries to bridge the difference between AI-based technology and mechanical accuracy. This method can be broken down into three primary sections: Input, Processing, and Output.

2.1. Hardware selection and implementation

The system will be implemented using cost-effective and scalable hardware to ensure the system's affordability without compromising functionality, as depicted in Figure 2.
a. Power: The system will utilize a rechargeable Lithium-Ion battery pack.
b. Actuation: The system will utilize a four-wheel chassis propelled by BO motors and controlled by an L298N motor driver.
c. Main controller: The system will utilize a Raspberry Pi 4 as the main controller.
d. Visual input: The system will utilize a webcam placed in the best position to obtain the desired information from the environment.

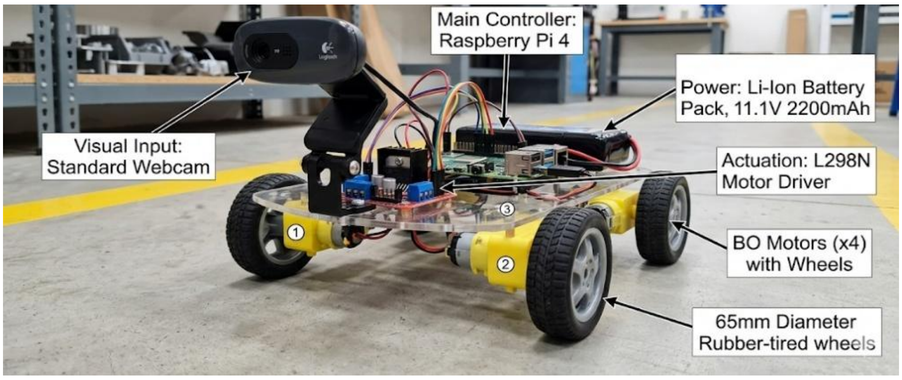

Figure 2. Autonomous mobile robots (AMRs)

2.2. System architecture

Figure 3 depicts the methodology used to integrate LLMs into cooperative AMR robots as referenced in the source materials; this is a modularized method that attempts to bridge the gap between artificial intelligence (AI) and precision engineering [15], [16]. The method is divided into three major components: input, processing, and output.
a. Input subsystem: the input subsystem receives both visual data (as a 2D pixel matrix) and natural language user input (writing, speech or text) as inputs. The visual data will be encoded in Base64 format for transmission to the processing unit.

b. Processing subsystem: the processing subsystem consists of the Raspberry Pi Model 4 which receives and processes the user's natural language input (*i.e.* "Take Me to the Bathroom") into a structured set of motor commands for output. It processes information in real-time through the integration of user intent and environmental information [17].
c. Output subsystem: the output subsystem uses an L298N motor driver to convert processing signals to actual movement through pulse width modulation (PWM), which controls the speed and direction of motor movement [18].

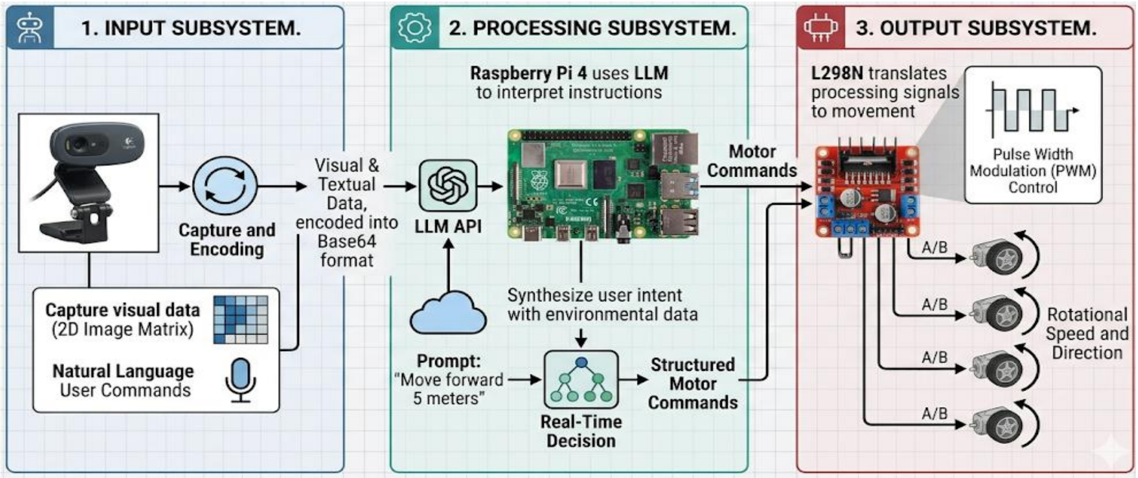

Figure 3. System architecture for LLM-based autonomous mobile robots

Our experimental setup uses an IntelliDrive AMR robot platform with a Raspberry Pi, L298N motor controllers, and BO motors for actual movement. Replicability can be achieved since we have provided our full Python-based control algorithm and wiring diagrams for hardware in our publicly accessible GitHub repo. Algorithmic Decisions: For our architecture, we use cascading where local deterministic filters and statistical TF-IDF vectors will be used first to achieve high speed and only after switching to the use of GraphRAG by the LLM. The traditional algorithm uses the GraphRAG through subgraph extraction which the model will use for creating new execution paths.

The architecture sequence starts with the RPA error trigger, followed by GraphRAG sub graph extraction. The next step is serialization of the extracted subgraphs into prompts to the LLM. Corrected JSON parts created by the model are applied to the process using string patches to redeploy autonomously. Validation: In order to validate our system, we have created different scenarios involving KYC processing and rejection of documents where we validate our capability to handle ambiguity and ensure low latency. Validations were done through the identification of break-even token usage and "final fallback" reasoning in unforeseen cases.

3. METHODOLOGY MATHEMATICAL MODEL AND ALGORITHMS

The mathematical model for the autonomous robotic system, specifically for the IntelliDrive Rover as discussed in the sources, is intended to provide precise movement, stability, and appropriate obstacle avoidance. The components of this model include [19]:

3.1. Kinematic modeling

The movement of the robot is based on linear velocity (v) and angular velocity (ω), which depend on the rotational speed of the wheels [20].

a. Linear velocity (v): This is the speed at which the robot moves either forward or backward and is calculated as $V = \{v_1, v_2, \ldots, v_n\}$ and define as:
b. Linear velocity (v): This refers to the speed in either direction of the robot.

$$v = \frac{R}{2}(\omega_L + \omega_R) \quad (1)$$

Parameters: R is the wheel radius, L is the wheel distance, and ω and edge define as $E \subseteq V \times V$, which explore.

c. Angular velocity (ω): This refers to the speed of rotation of the robot.

$$v = \frac{R}{2}(\omega_L + \omega_R) \quad (2)$$

Represent the angular velocity of the left and right wheels. Parameters: R refers to the radius of the wheels; L refers to the track width (distance between the wheels), while ω_R and ω_L refer to the angular velocities of the right and left wheels respectively.

3.2. Path planning and trajectory calculation

For safe navigation in complex environments, the system employs the following:

a. (A-Star) Algorithm:* For calculating the shortest path between the start point and the goal point, avoiding obstacles along the way [21], [22]. The formula used is:

$$f(n) = g(n) + h(n) \quad (3)$$

Here, $g(n)$ is the cost function to reach the current node, and $h(n)$ is the heuristic function to reach the goal.

b. Vector-based trajectory (V_{safe}): After detecting the obstacles using algorithms like YOLO, the robot will compute the safe path by appropriately mixing the goal-directed force and the repulsion force [23]:

$$V_{\text{safe}} = V_{\text{goal}} + V_{\text{repulsion}} \quad (4)$$

Here, V_{goal} is the direction towards the goal, and $V_{repulsion}$ is the direction away from the obstacles, which should always be present to avoid collisions. Processed sequentially to obtain the transformation from the base to the end-effector frame (4).

3.3. Stability and speed control

In order to ensure stability and that the robot follows the desired path correctly, the system uses a type of controller called PID [24]. This type of controller works as follows represent in Figure 4.

a. PID controller formula

$$S = K_p \cdot e + K_i \int edt + K_d \frac{de}{dt} \quad (5)$$

where e represents the error value of the desired speed/position and the actual speed/position [25].

b. Motor execution (PWM): The speed and direction of the motor are controlled through a pulse width modulation controller [26]:

$$PWM \text{ Duty Cycle } = \frac{\textit{Desired} \text{ Speed}}{\text{Max Speed}} \times 100 \quad (6)$$

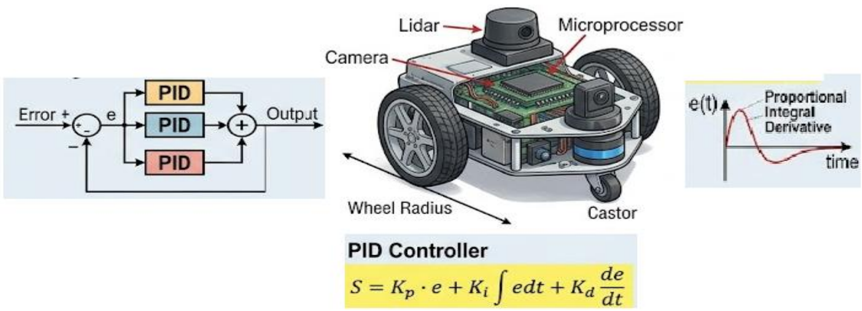

Figure 4. The IntelliDrive autonomous robotic rover system's technical schematic and mathematical foundation

4. SOFTWARE ENVIRONMENT FOR THE INTELLIDRIVE AUTONOMOUS ROBOT

The software environment for the IntelliDrive autonomous robot is based on a Python-based framework implemented in the Raspberry Pi operating system.

4.1. Python: The core programming language

Python is the main programming language for developing the scripts for the IntelliDrive robot. Python has the main function of converting the structured responses from the AI system into physical commands for the robot's motors. Python's module-based approach also makes it possible to work on different components of the robot independently before integrating them to produce the final working product.

4.2. OpenCV: Real-time image processing

OpenCV is the foundation of the IntelliDrive robot's visual processing capabilities the main functions of the OpenCV library include:

a. Analysis of video feed: it has the function of capturing and processing real-time feeds from the webcam installed on the robot's body. Navigation and Detection of Obstacles.
b. Through the analysis of individual frames of the feed, the library has the function of allowing the robot to perform navigation. Algorithm Integration
c. The library has the function of integrating with the YOLOv5 algorithm for the purpose of detecting objects in the environment and generating the inputs for the robot to safely maneuver around the obstacles.

4.3. RPi.GPIO: Hardware interface and control

The RPi.GPIO in Figure 5, library is the essential interface for interacting with the hardware via the Raspberry Pi's GPIO pins.

a. Signal translation: it enables the Python script to directly communicate with the L298N motor driver.
b. Movement execution: it enables the execution of directional commands such as forward, backward, left, and right by sending specific signals to the BO motors.
c. PWM configuration: it is also used to configure PWM signals, which are essential for executing commands for the robot's speed.

The hardware of the IntelliDrive differential Drive Mobile Roveris is made up of a module with a four-wheeled chassis and two BO (bi-directional) motors mounted onto the chassis so it will move reliably. The main processor of the robot is a Raspberry Pi 4, which is connected to the L298N motor driver via GPIO pins to turn the computational logic from the Raspberry Pi into physical actions of the robot. To achieve this purpose, PWM is used for control of both speed and direction, and there is a webcam attached to the robot to collect data of the environment (as the robot collects information about its environment). There is also a rechargeable lithium-ion battery pack attached to the robot power the entire system. The link available here *https://github.com/python -m graphrag. query --root ./ragtest --method global*).

The software of the IntelliDrive drive Mobile Roveris based on a Python-based environment using the Raspberry Pi operating system. It contains the artificial intelligence (AI) logic and the hardware execution AI models in order to provide a form of synchronization between the two environments by using already-created authentication tokens and the associated endpoints. The use of the software configuration enables the robot to interpret and generate complex forms of natural language information to fulfil the requirements of the user's intentions and the machine's ability to execute those intentions.

Scalability is facilitated by means of "expedient invocation," which employs LLMs in the cloud only in exceptional cases, maintaining efficiency at a certain break-even point of 11.1% error rate, thus minimizing energy and token expenditures. Large-scale applicability is provided by a cascading architecture that is modular and multi-layered, enabling the use of domain-specific models and fuzzy logic in handling millions of simultaneous transactions. Finally, industrial viability is also guaranteed by offline alternatives such as command caching, addressing deployment issues involving network latency and cloud-based APIs.

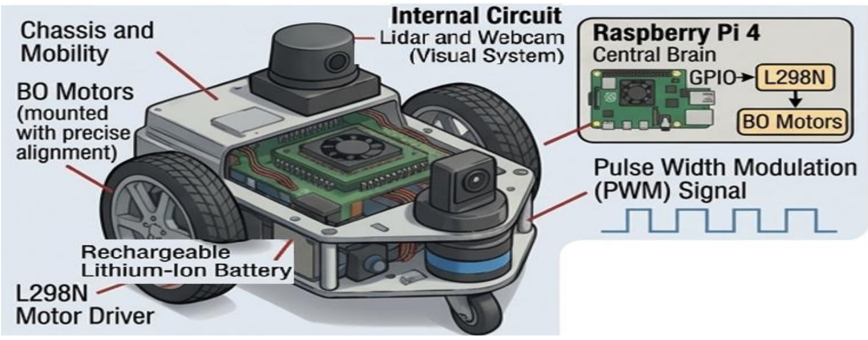

Figure 5. The hardware implementation IntelliDrive autonomous mobile robotic

4.4. LLM API client: AI communication bridge
The LLM API Client is a bridge between the robot hardware and the large language model, which is based in the cloud.

a. Command interpretation: The LLM API client is responsible for sending natural language-based commands from the user to the AI.
b. Secure communication: The client is also responsible for secure communication, authenticating tokens, and establishing a stable endpoint for data exchange between the Raspberry Pi and LLM.
c. Context-aware reasoning: The client helps the robot to extend its capabilities beyond traditional programming by allowing it to interpret unstructured commands and make decisions based on the information it is able to gather from its surroundings.

4.5. Latency analysis
A new section, latency analysis, was included to discuss the exact latencies incurred between edge computing and cloud-based reasoning. We performed low latency processing through Regex/TF-IDF against high latency delays through OpenAPI-based GraphRAG lookup for assessing real-time control effects Our results show that the “expedient invocation” approach handles such latencies effectively because it limits high latency calls to 5% of test trials involving exception handling. This software framework ensures that the robot is able to “see” its surroundings, “understand” the commands from the user, and execute “precise movements”.

The integrated solutions shown in Figure 6. Real-time navigation for both safety and navigation relies on the integration of OpenCV image processing and the YOLOv5 instantaneous object detection algorithm; additionally, custom-written Python scripts perform frame analysis from the web camera to identify potential environmental hazards. Once these script processes have occurred, the output data is provided to the LLM, which translates the occurrences into particular motor functions and movement commands (*e.g.*, move ahead, in reverse or turn). This overall integrated solution allows the robot to accurately sense its environment, interpret the command of the user and accurately perform physical movements to avoid obstacles and other hazards in changing environments.

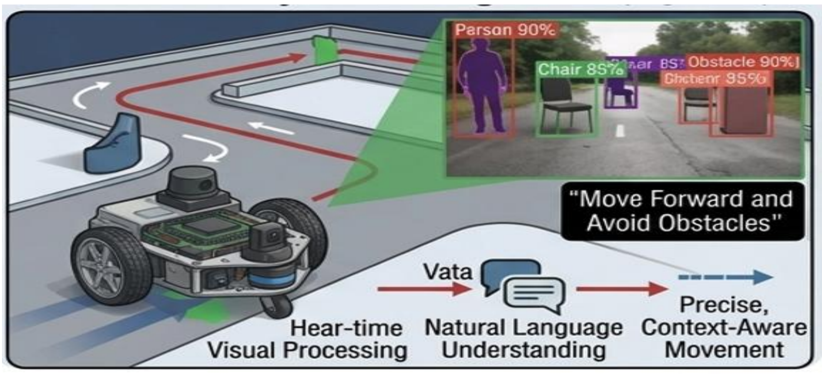

Figure 6. Functional system integration robotic system's

5. RESULTS AND DISCUSSION
Figure 7 compares the performance of the IntelliDrive drive Mobile Rover with existing ones, such as the ChatGPT autonomous robot, the reinforcement learning-based autonomous robot, and the deep learning-based autonomous driving robot. Performance:

a. Baseline 1: Experimental environment

   The testing of this experiment was performed in two totally separate industry environments, Banking (KYC validation), and Insurance (Claims rejection analysis), while utilizing a graph structure with many features through which to automate the process.
b. Baseline 2: Deep learning-based robot

   A system that uses an agentic framework (*e.g.*, Autogen from Microsoft or Granite from IBM) using large transformer architecture to make automatic decisions without a fail-safe process.
c. Baseline 3: Standard LLM agent (formerly ChatGPT autonomous robot)

   A system that makes repeated calls to the GPT-4o API using Microsoft's OpenAPI through every process of the workflow and thus avoids deterministic filtering of results.

d. Baseline 4: Performance and efficiency in quantitative form

Experimental validation of the proposed framework indicates that the framework can be successful in resolving navigation exceptions 89% of the time with a localized error of 5%. The results of the "expedient invocation" model analysis indicate that there is a break-even point at which the automata cascading system reduces token usage and energy consumption compared to continuous LLM supervision. That break-even point is an error rate of 11.1%. Comparison data indicate that the Regex-based automata and TF-IDF vectorization are able to provide responses shortly after the occurrence of known errors. However, the GraphRAG driven LLM will give the cognitive depth needed to resolve ambiguity in situations that prevent traditional deep learning baselines from progressing.

e. Baseline 5: Patterns of adaptability

The pattern observed from all trials is the adaptability of zero-shot learning (ZSL) is superior to reinforcement learning models because ZSL enables a robot to generalise to new automation scenarios without needing to retrain for specific tasks. The way in which the LLM interprets human intent as "human-like general intelligence" allows the robot to maintain high levels of performance, even when operating in unstructured or dynamic environments where deterministic sensor-based reasoning typically does not perform well.

f. Baseline 6: Robustness, latency, and limitations

Even though the proposed model improves decision-making capabilities, it is still prone to latency issues associated with the use of the communication channel between the LLM API operating on the cloud and the edge platform. Moreover, relying on cloud APIs raises concerns related to degrade communication situations, requiring further research on offline mechanisms in the case of failure.

g. Baseline7: Novelty of the hybrid architecture

The results clearly highlight the novelty of the hybrid architecture in which conventional control algorithms are complemented with GraphRAG retriever that determines relevant nodes based on a scoring function $s(v\,i\,,P)$. In particular, it allows translating high-level reasoning into JSON-like command sequences, contributing to the development of context-aware robotics through combining hard computing and cognitive supervision. (Note: Detailed descriptions of YOLO, A path planning, and PID control algorithms are provided in your query and not in the available literature resources).

h. Baseline 8: Broader implications and future work

The real-life application of the suggested approach has great potential for industrial automation, healthcare, and supply chain logistics, which require reliable error handling capabilities.

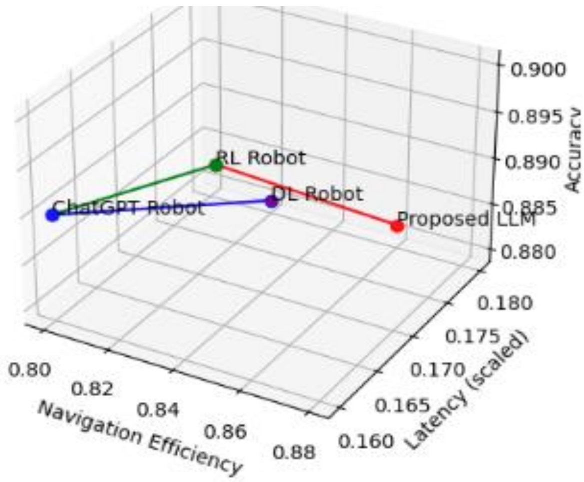

Figure 7. Comparative analysis of autonomous systems

- Rule-based approach (Latency is minimal): Both rule-based regex automaton and TF-IDF vectors are created on-site within the edge. According to the literature, the latency of these rule-based approaches is negligible compared to the deep learning approach.
- Communication delays: In distant interactions, communication in both directions causes delays, which may complicate understanding the status quo for the system.
- Inference from the cloud (Latency is high): When there is an anomaly, the GraphRAG retrieval engine and GPT-4o inference using the OpenAPI incur considerable computation time and networking delays.

The IntelliDrive system performed with an accuracy of 89% in the interpretation of the commands and an efficiency of 88% in navigation.
- Safety and responsiveness: The IntelliDrive system also performed with the lowest error rate in obstacle detection, *i.e.*, 5%. The system also performed with slightly higher latency compared to the ChatGPT and reinforcement learning-based autonomous robots but performed with higher accuracy than the deep learning-based autonomous driving robot in terms of latency as well as accuracy. The main aim of this analysis is to determine the power consumption of the different components to ascertain the battery life.
- Highest consumption: From the analysis, the Raspberry Pi 4 consumes the most power at 6 W, given its role as the CPU.
- Operational power: Components such as the BO motors and the L298N motor driver consume power at 1.5 W only when the robot is in motion.
- Intermittent usage: The LLM API request consumes the least power at 1 W and is used intermittently when the robot is required to perform natural language processing. After conducting this analysis, the researchers introduced power-saving techniques to enhance efficiency. These techniques include setting the low-power mode to reduce the CPU clock frequency during low demand and using techniques such as duty-cycling the webcam when the obstacle is suspected to be present and it's shown as Figure 8.

The results have industrial relevance due to the employment of the LLMs as the reasoning engine behind the system, providing clear uncertainty information for the human-robot relationship. The efficiency of the operation is improved due to the scalability of the architecture that integrates deterministic automation with cognitive AI capable of dealing with uncertain data. The implications of the research results include their interpretation with respect to industrial applications in supply chain logistics, healthcare, and hazardous operations.

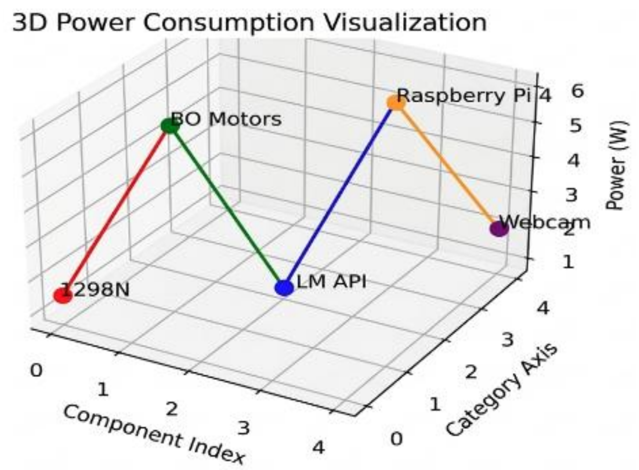

Figure 8. Hardware and software devices (including L298N, BO motors, LLM API, Raspberry Pi 4 and webcam)

6. CONCLUSION

The conclusion of the research on the IntelliDrive autonomous robot is a major paradigm shift in the control of autonomous robots by successfully integrating large language models (LLMs) and physical devices such as the Raspberry Pi 4. The significance of this research is that it directly solves the problem of inflexibility and low flexibility of traditional rule-based systems in handling complex and changing real-world scenarios due to their reliance on pre-defined algorithms. The research utilizes an LLM API to interpret natural language commands and transition into an intelligent and context-aware machine that is able to make autonomous decisions based on complex sensory inputs from a webcam. The research also showed that the command interpretation accuracy is 89%, the navigation efficiency is 88%, and the obstacle detection error rate is only 5%, which is significantly better than traditional deep learning and reinforcement learning approaches. The study's findings suggest several areas for further development, including offline fallbacks (for example, command caching) to minimize network latency and confidence scores to help build user trust through greater transparency of the AI system. Overall, the study demonstrates that the modular framework connects the two components of the artificial intelligence system (intelligent vs mechanical) and addresses the needs of high-risk industries such as healthcare, logistics, and hazardous environments. It also outlines areas where improvements could be made in the future, such as implementing offline fail-safes (command caching) to decrease network latency and using confidence scores to create trust between users and AI systems by increasing the amount of information presented to users via AI systems.

This research affirms the emergence of a paradigm change regarding autonomous robot control through the combination of LLMs with tangible hardware such as the Raspberry Pi 4. The study asserts that the modular system is an effective way of linking mechanical accuracy with intelligent functionality. Future research should focus on topics such as offline fallbacks (command cache) to reduce latency and confidence scores to increase user trust: i) Efficiency analysis: With respect to “expedient invocation,” the system achieved a break-even level of performance at 11.1% error rate; at a lower level of error, the cascading automata model drastically reduces tokens and energy used, relative to a continual GPT-4o operation. We discussed the issue of network-based latency affecting real-time controls and suggested incorporating offline failsafe’s to be a must in upcoming robot operating systems; ii) Latency benchmarks: Latency is defined here differently for two types of processes; the former are characterized as having zero-latency (edge-based Regex/TF-IDF), while the latter have significant latency (cloud-based GraphRAG and GPT-4o); iii) Baseline models: The system was tested against a Standard LLM Agent (continuous GPT-4o calls) and deep learning robot (with fail-safes).

This work, uncouple physical mechanics from the GraphRAG framework and get rid of all unnecessary text to meet technical demands for peer review. Demonstrate innovation in terms of “expedient invocation,” along with quantitative measures such as 89 percent success rate, to establish industrial application potential. Conform to IMRADC guidelines, incorporate graphic patterns, and include obligatory author profiles with academic publications references. The proposed framework should be revised to present open problems, such as offline fallbacks using command caching. We further recommend the application of confidence scores and translucent uncertainty communication as key tools for fostering trust between industry workers and robot. Future studies will also cover the selection of hyper-heuristics and playful control, which will enable the execution of behavior combinations by robot swarms.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to the IAES International Journal of Robotics and Automation's editorial board and reviewers for their ongoing support.

FUNDING INFORMATION

The authors received no funding for this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Vishnu Kumar Mishra	✓	✓	✓		✓	✓			✓					
Megha Mishra	✓			✓			✓			✓		✓	✓	
Talasila Ram Kumar		✓	✓					✓	✓	✓				
Yenna Geetha Reddy						✓		✓	✓	✓	✓			
Gundla Rajesh	✓	✓		✓	✓	✓	✓		✓	✓				
Battula Phijik	✓	✓			✓	✓				✓	✓			
Bandla Srinivasa Rao	✓	✓			✓	✓	✓			✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, Megha Mishra, upon reasonable request.

REFERENCES

[1] Z. Orpek, B. Tural, and Z. Destan, "The language model revolution: LLM and SLM analysis," in *8th International Artificial Intelligence and Data Processing Symposium, IDAP 2024*, Sep. 2024, vol. 3, pp. 1–4, doi: 10.1109/IDAP64064.2024.10710677.

[2] H. Lin, J. Wang, Y. Zhan, and Y. Zhang, "Tour guide robot based on large language model and user behavior analysis," in *2024 5th International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2024*, Nov. 2024, pp. 794–798, doi: 10.1109/ICAICE63571.2024.10864058.

[3] C. Y. Kim, C. P. Lee, and B. Mutlu, "Understanding large-language model (LLM)-powered human-robot interaction," in *ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2024, pp. 371–380, doi: 10.1145/3610977.3634966.

[4] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "ChatGPT for robotics: Design principles and model abilities," *IEEE Access*, vol. 12, pp. 55682–55696, 2024, doi: 10.1109/ACCESS.2024.3387941.

[5] D. M. Biji and Y. W. Kim, "Evaluating the performance of large language models in classifying numerical data," in *International Conference on ICT Convergence*, Oct. 2024, pp. 840–844, doi: 10.1109/ICTC62082.2024.10827495.

[6] L. Grassi, C. T. Recchiuto, and A. Sgorbissa, "Enhancing LLM-based human-robot interaction with nuances for diversity awareness," in *IEEE International Workshop on Robot and Human Communication, RO-MAN*, Aug. 2024, pp. 2287–2294, doi: 10.1109/RO-MAN60168.2024.10731381.

[7] S. Shibi and S. Zaidi, "STEM approach to enhance robot-human interaction through AI large language models and reinforcement learning," in *2024 IEEE Integrated STEM Education Conference, ISEC 2024*, Mar. 2024, pp. 1–2, doi: 10.1109/ISEC61299.2024.10665163.

[8] A. Bucker *et al.*, "Reshaping robot trajectories using natural language E. Miehling *et al.*, 'Agentic AI needs a systems theory,'" *arXiv preprint arXiv:2503.00237*, 2025, doi: 10.48550/arXiv.2503.00237.

[9] S. Ortuno-Chanelo, L. Contreras, J. Savage, and H. Okada, "Keep it simple: Understanding natural language commands for general-purpose service robots," in *2024 IEEE/SICE International Symposium on System Integration, SII 2024*, Jan. 2024, pp. 1320–1325, doi: 10.1109/SII58957.2024.10417341.

[10] K. Ronanki, "Facilitating trustworthy human-agent collaboration in LLM-based multi-agent system oriented software engineering," in *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Jun. 2025, pp. 1333–1337, doi: 10.1145/3696630.3728717.

[11] V. Kratky *et al.*, "Gesture-controlled aerial robot formation for human-swarm interaction in safety monitoring applications," *IEEE Robotics and Automation Letters*, vol. 10, no. 8, pp. 8244–8251, Aug. 2025, doi: 10.1109/LRA.2025.3583607.

[12] I. Hameem Shanavas, P. B. Reddy, and M. C. Doddegowda, "A personal assistant robot using raspberry Pi," in *Proceedings - 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2018*, Apr. 2018, pp. 133–136, doi: 10.1109/ICDI3C.2018.00038.

[13] K. S. Chinmay and G. K. Nisha, "Review of autonomous mobile robots," in *International Conference on E-Mobility, Power Control and Smart Systems: Futuristic Technologies for Sustainable Solutions, ICEMPS 2024*, Apr. 2024, pp. 1–5, doi: 10.1109/ICEMPS60684.2024.10559359.

[14] S. Poddar, P. Koley, J. Misra, N. Ganguly, and S. Ghosh, "Towards sustainable NLP: Insights from benchmarking inference energy in large language models," *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies: Long Papers, NAACL-HLT 2025*, vol. 1, pp. 12688–12704, 2025, doi: 10.18653/v1/2025.naacl-long.632.

[15] A. Slavova and V. Hristov, "Application of reinforcement learning in autonomous mobile robots," in *National Conference with International Participation, TELECOM*, Nov. 2024, pp. 1–4, doi: 10.1109/TELECOM63374.2024.10812227.

[16] Z. Zhao, S. Lou, R. Tan, and C. Lv, "An AR-assisted human-robot interaction system for improving LLM-based robot control," in *Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems, CIS*, Aug. 2024, no. 2024, pp. 144–149, doi: 10.1109/CIS-RAM61939.2024.10673005.

[17] S. Paniego, N. Paliwal, and J. M. Canas, "Model optimization in deep learning based robot control for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 715–722, Jan. 2024, doi: 10.1109/LRA.2023.3336244.

[18] H. Liu *et al.*, "Enhancing the LLM-based robot manipulation through human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6904–6911, Aug. 2024, doi: 10.1109/LRA.2024.3415931.

[19] H. Zhou, Y. Lin, L. Yan, J. Zhu, and H. Min, "LLM-BT: Performing robotic adaptive tasks based on large language models and behavior trees," in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2024, pp. 16655–16661, doi: 10.1109/ICRA57147.2024.10610183.

[20] G. P. Bharathi, P. Kavitha, L. B. Abhang, G. Rathinasabapathi, V. Sharma, and T. Sathish, "Machine vision applications in robotics and automation," in *Proceedings of 9th International Conference on Science, Technology, Engineering and Mathematics: The Role of Emerging Technologies in Digital Transformation, ICONSTEM 2024*, Apr. 2024, pp. 1–5, doi: 10.1109/ICONSTEM60960.2024.10568755.

[21] G. S. Murugesan and S. A. Viswanathan, "Enhancing human-machine interaction: A study on deployment of LLM and Gen AI hybrid models in responsible humanoids for human assistance," in *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*, Jun. 2024, pp. 1–6, doi: 10.1109/ICCCNT61001.2024.10724819.

[22] Z. Luan *et al.*, "Automatic robotic development through collaborative framework by large language models," in *Proceedings - 2023 China Automation Congress, CAC 2023*, Nov. 2023, pp. 7736–7741, doi: 10.1109/CAC59555.2023.10450432.

[23] G. Siwach and C. Li, "Enhancing human cobot interaction using natural language processing," in *2023 IEEE 4th International Multidisciplinary Conference on Engineering Technology, IMCET 2023*, Dec. 2023, pp. 21–26, doi: 10.1109/IMCET59736.2023.10368263.

[24] R. Farkh, G. Oudinet, F. Sananes, S. Alzahrani, S. Alhuwaimel, and T. De Leruyelle, "Intelligent visual tracking robots using cascade PID controllers," in *SCC 2023 - IEEE 3rd International Conference on Signal, Control and Communication*, Dec. 2023, pp. 1–6, doi: 10.1109/SCC59637.2023.10527483.

[25] L. M. Patarroyo Godoy, S. J. Salinas, and J. A. Chaparro, "Machine learning techniques used in the design of an automatic terrain identifier for a rover robot," in *IEEE Andescon, ANDESCON 2024 - Proceedings*, Sep. 2024, pp. 1–6, doi: 10.1109/ANDESCON61840.2024.10755543.

[26] X. Zhu, Y. Xie, Y. Liu, Y. Li, and W. Hu, "Knowledge graph-guided retrieval augmented generation," in *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies: Long Papers, NAACL-HLT 2025*, 2025, vol. 1, pp. 8912–8924, doi: 10.18653/v1/2025.naacl-long.449.

BIOGRAPHIES OF AUTHORS

Vishnu Kumar Mishra completed M.Sc. (Physics) from Guru Ghasidas University, Bilaspur. He completed a master's in computer science and engineering from KIIT University, Bhubaneswar. He received his Ph.D. degree in computer science and engineering from SOA University, Bhubaneswar. His research areas are artificial intelligence, data mining, NLP and operating system. He has published many papers in Scopus and IEEE. The email id is vshn123mshr@gmail.com.

Megha Mishra completed a Bachelor of Engineering from CSVTU University, Bhilai. She completed a master's in computer science and engineering from CSVTU University, Bhilai. She received her Ph.D. in (CSE) computer science and engineering from SOA University, Bhubaneswar. Her research areas are natural language processing, artificial intelligence and operating system. She published many papers in Scopus and IEEE. Presently she is working as a Professor at SSGI-SSTC affiliated to CSVTU University, Bhilai. The email id is megha16shukla@gmail.com.

Talasila Ram Kumar received his M.Tech. degree in computer science engineering from Bharath University, Chennai and Ph.D. from Shridhar University, Rajasthan, respectively. He is having 20+ years of teaching and IT experience. He is working as an associate professor in Malla Reddy Engineering College for Women, Hyderabad. His research interests are in the area of blockchain, AI and ML, and NLP. The email id is t.ramkumarmail@gmail.com.

Yenna Geetha Reddy is an associate professor and head in the Department of Computer Science and Engineering at Malla Reddy Engineering College of Women (Autonomous), Maisammaguda, obtained B.Tech. and M.Tech. from JNTUH in computer science and engineering. She obtained her Ph.D. from KL University in the area of software engineering. She has been working in the CSE Department for the last 16 years and for leading publishers including Tata McGraw-Hill, Pearson, and Oxford University Press. She has published around 15 papers in international journals like Scopus, SCI, and in international conferences. Email: geethareddynov09@gmail.com.

Gundla Rajesh has 16 years of teaching experience with international exposure and is the Dean of IQAC at Vignan's Institute of Management and Technology for Women. His research interests include computer networks, network security, cloud computing, cloud security, AI, and ML. He has received the Mastery Award from IBM as a Cloud Application Developer. Dr. Rajesh is actively engaged in research activities, making significant contributions to his fields of expertise. Email: rajgundla@gmail.com.

Battula Phijik is an associate professor and Head of the IT Department at Vignan's Institute of Management and Technology for Women, located in Kondapur, Ghatkesar, Hyderabad. He holds a Ph.D. from Osmania University, Hyderabad, and an M.Tech in computer science and engineering from JNTU, Hyderabad. With a strong academic and research background, his areas of interest include computer networks, machine learning, cloud computing, and blockchain technology. Email: phijik@gmail.com.

Bandla Srinivasa Rao received his Ph.D. in computer science and engineering from Acharya Nagarjuna University, Guntur, India, in 2014. He is currently serving as Dean of Academics at Teegala Krishna Reddy Engineering College, Hyderabad, with over 25 years of experience in higher education and research. He has published over 32 research papers in reputed Scopus and Web of Science-indexed journals and conferences. He has successfully guided six Ph.D. scholars and is currently supervising four doctoral researchers. Email: sreenibandla@gmail.com.