Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

ROB-AI: An Offline Voice-Controlled Smart Home System with Context-Aware On-Device Intent Reasoning

Athishkirthik J D.[1], Pranav Y.[2], Sridevi S.[3]

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India.

E-mail: [1]aj2075@srmist.edu.in, [2]py1378@srmist.edu.in, [3]sridevis2@srmist.edu.in

Orcid ID: [1]0009-0008-3992-4928, [2]0009-0008-4566-902X, [3]0000-0003-0471-8653

Abstract

This work introduces Responsive-Operational-Bot (ROB-AI), a localized smart home automation system designed to function entirely offline, interpreting natural language voice commands without dependence on external cloud. The proposed architecture seamlessly combines speech recognition on the device, a hybrid intent detection engine, and RESTful IoT-based device control over a local network. A lightweight, context-aware reasoning layer is added to connect human language with strict machine instructions. This lets the system correctly understand indirect or incomplete commands by checking the current state of connected hardware. The system is designed to minimize latency significantly, remove the use of all external internet connections, and protect user data. Experimental results show that ROB-AI works very well in the real world, with standard commands running almost instantly and maintaining a high accuracy rate of 92–96% for complex semantic processing. The variation in the accuracy percentage is observed in cases such as noisy environment and high-speed command input. In the future, to scale up the IoT end points and to enable seamless expansion of web-based dashboard interfaces, a modular architecture is effectively designed.

Keywords: Smart Home, Voice Control, Internet of Things (IoT), Edge Computing, Offline AI, Automation.

DOI: https://doi.org/10.36548/jucct.2026.3.001
Received: 03.04.2026, received in revised form: 04.05.2026, accepted: 26.05.2026, published: 05.06.2026

1. Introduction

Smart home automation is one of the fastest growing fields of consumer technology, largely to the explosive growth of the IoT over the last decade, along with rapid advancements in Artificial Intelligence (AI). The use of voice-activated systems as the preferred method for people to interact with computers at home is due to their ease-of-use (no physical interaction necessary). But most of the commercial solutions that are currently popular have a major architectural flaw: they depend too much on cloud-based online processing. In these traditional systems, a person's voice data is recorded, packaged, and sent to remote server farms for transcription and intent extraction.

Then, a control signal is sent back to the local device. This method provides access to substantial computational resources; however, it also introduces functional and operational limitations. It naturally causes latency in the network, needs a stable and permanent internet connection, and most importantly, it raises serious privacy issues. Sending continuous audio data from a private home to outside corporate servers is a security risk that many users are becoming less willing to take.

This paper presents ROB-AI, a robust, completely offline voice-controlled automation ecosystem that manages the entire computational pipeline from audio transcription to hardware execution locally at the edge, directly addressing these limitations.

1.1 System Motivation

The primary motivation driving the development of ROB-AI is the decentralization of smart home logic. There are many times when a user may have inconsistent access to the Internet or experience a slower connection due to bandwidth throttling in real world examples, but by eliminating dependency on any outside Cloud infrastructure will allow a local system to continue to function regardless of the outside network state. Also, processing localized audio using edge hardware nearly eliminates any type of network round trip delays which will provide for a faster and more responsive user experience. Most importantly, keeping all data lifecycle within the boundaries of the local Wi-Fi network will ensure absolute user trust by ensuring that private conversations never leave their home’s physical footprint.

2. Related Work

Smart home automation has evolved from traditional rule-based control systems to intelligent environments capable of understanding user intent and adapting to contextual information. Early smart home implementations primarily relied on predefined commands and fixed interaction patterns, limiting flexibility and natural user interaction. Controller-based automation systems provided reliable monitoring and control of household appliances but offered limited support for contextual reasoning and intelligent decision-making [3].

Recent research has focused on integrating edge computing and artificial intelligence into smart home environments. Edge computing does its processing near the actual devices, shaving down latency issues, keeping everything dependable, and cutting back on that heavy reliance on cloud servers [4]. Then there are cloud-to-edge setups which boost functionality in homes, all while preserving privacy and ensuring snappy performance [5]. Progress in local smarts has also beefed up how well assistants can guess what users intend. Models using ontologies along with smaller LLMS work their magic by inferring intentions reliably and all within limits of resource constraints [1].

Context awareness has become a critical component of modern smart home systems. Research has shown that incorporating environmental conditions and device-state information significantly improves the interpretation of ambiguous or incomplete user commands while maintaining low power consumption [8]. These capabilities enable more natural interactions and support decision-making beyond simple command execution. Speech recognition technology has also matured significantly, making offline voice processing feasible for edge devices.

Deep neural network-based automatic speech recognition (ASR) systems have achieved substantial improvements in recognition accuracy and robustness across diverse operating conditions [6]. More recent studies demonstrate that real-time speech-to-text processing can be performed directly on edge devices with minimal latency, enabling responsive voice-controlled applications without cloud dependency [7]. On the hardware side, low-cost microcontrollers and IoT platforms have accelerated the deployment of smart home solutions.

ESP8266-based automation systems provide an affordable and scalable foundation for device control and monitoring through local wireless networks [9]. IoT-based home automation

architectures further demonstrate the effectiveness of integrating sensors, controllers, and communication modules to achieve real-time appliance management and monitoring [10].

Despite these advances, many existing solutions continue to rely either on cloud-based processing or computationally intensive architectures that increase deployment complexity and cost. ROB-AI addresses these limitations by combining offline speech recognition, hybrid intent recognition, context-aware reasoning, and ESP8266-based actuation within a fully localized architecture. This approach aims to provide a scalable, privacy-preserving, and cost-effective smart home automation platform while maintaining responsive real-time performance [1], [4], [8], [9].

3. System Architecture

The ROB-AI ecosystem provides optimal performance without requiring cloud resource management as a result of its modular, decoupled architecture (Figure 1). The ecosystem consists of Three (3) major layers of functionality, namely: Input Layer, Intent Processing Layer, and Actuation Layer.

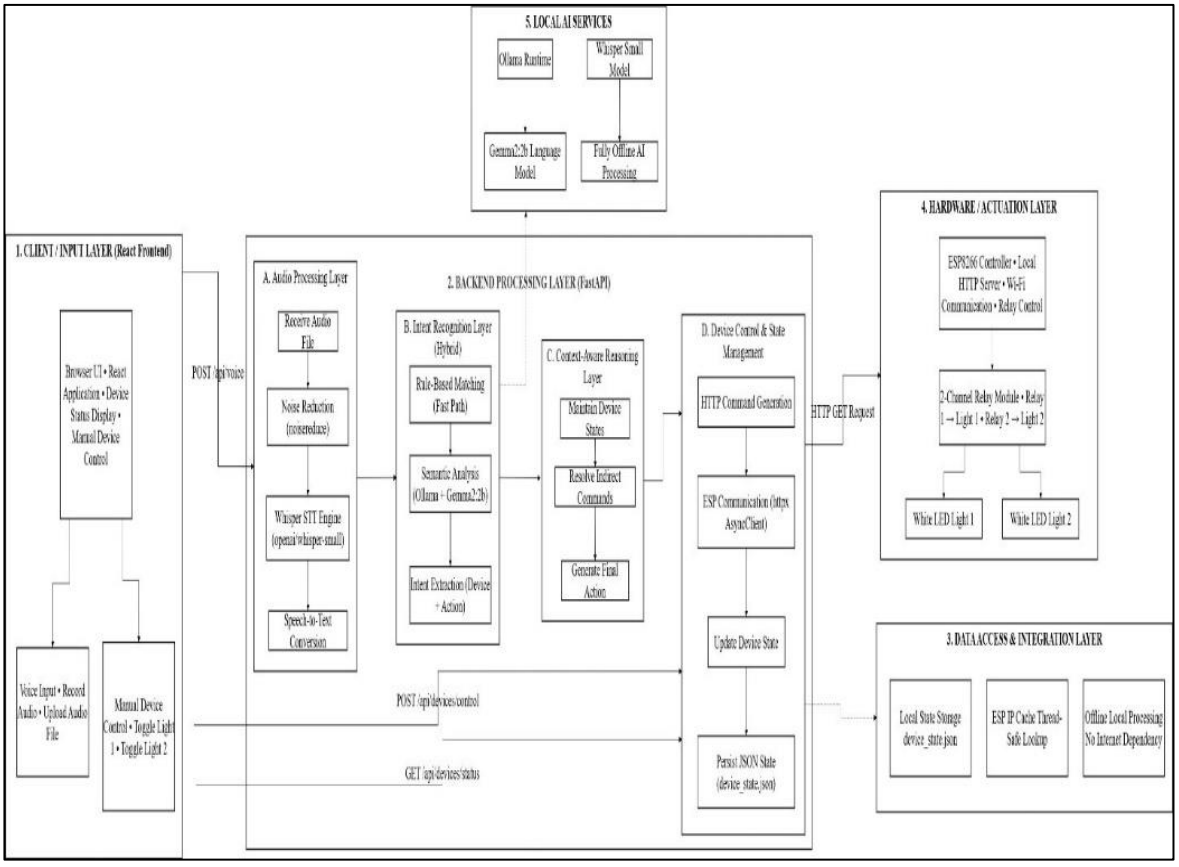

Figure 1. Overall Architecture of the Proposed ROB-AI System

3.1 Input Layer

The input layer acts as the system's sensory interface, capturing human voice commands through a localized microphone array. Noise reduction and audio formatting techniques, like converting raw web audio blobs into standardized, 16kHz mono WAV streams, are used at this stage to clean the signal. This ensures high-quality audio is sent to the transcription engine, no matter the background noise.

3.2 Intent-Processing Layer

The cognitive processing area of ROB-AI is housed in this processing layer. It is located on a local host server and is comprised of three totally different local subsystems:

- Offline Speech-to-Text (STT): The purpose of this is to convert of cleaned-up audio signals into raw text.
- Hybrid Intent Recognition: This will process the generated text from the STT to determine which type of device will be executed on and determine what type of action will be performed on that device.
- Context-Aware Reasoning Module: This will evaluate the intent generated from the Hybrid Intent Recognition in relation to the state of the physical hardware in the real world and create a logical conclusion about what action should occur.

3.3 Actuation Layer

The execution layer acts as a bridge between the digital representation of the intent (AI) and its actual execution within the physical world through machine-to-machine communication over a localized Wi-Fi network via standard HTTP protocols among the Internet of Things (IoT) microcontrollers. This independent communications layer from the electrical execution layer allows for greater scalability and easier maintenance of the AI process versus the pure electrical execution.

4. Proposed System

The key innovation of ROB-AI is its approach to interpreting language and executing commands without the enormous neural networks typically associated with cloud-based architectures.

4.1 Voice Processing

Once the user has spoken, ROB-AI captures the voice data (speech) and processes it through an offline Speech Recognition (SR) system using Edge optimized STT (Speech to Text) libraries. The end result is a timely, highly accurate textual representation of the speech (sub-second response; no latency added by cloud API handshake).

4.2 Intent Recognition

ROB-AI processes voice inputs using a hybrid intent recognition framework that combines rule-based command matching with natural language understanding techniques. The backend is implemented using Python and FastAPI. User speech is first converted into text using the Whisper speech recognition model operating locally on the host device.

Intent identification follows a two-stage process. Initially, the system attempts to match the transcribed text against a predefined command repository containing frequently used commands such as:

- Turn on Light 1
- Turn off Light 2
- Switch on all lights

This rule-based approach provides fast execution with minimal computational overhead. When a command cannot be matched using predefined rules, the text is forwarded to the Gemma2:2b language model running locally through the Ollama framework for semantic analysis.

The intent recognition workflow consists of:

- Audio preprocessing and noise reduction
- Speech-to-text conversion using Whisper
- Rule-based command matching
- Semantic intent analysis using Gemma2:2b
- Device and action identification

- Generation of HTTP control commands

This hybrid architecture enables efficient interpretation of both structured and natural-language commands while maintaining low latency and offline operation.

4.3 Context-Aware Reasoning

The machine's understanding of ordinary language is very complex because it is often very context-sensitive. People who provide input into a computerized device typically do not talk to them in machine terms. The ROB-AI reasoning layer saves a continuous memory of the current state of all devices that are connected. This allows the system to have the ability to interpret ambiguous commands. When a user says, "Turn it off," the system looks at its internal state to see what the active device is so that it can target that device. Likewise, when a user says, "Make it dark in here," the system evaluates its current lighting state and uses its own judgment to determine the lights that are currently in an active state and to send a toggle command to turn those lights off.

Algorithm: Context Aware Reasoning

Input : User Speech Text, Device State
Output : Executable Command

BEGIN
Receive the user's speech input.
Convert the speech into text format.
Convert all text to lowercase.
Remove irrelevant words and unnecessary phrases.
Correct common speech recognition errors.
Analyze the sentence to determine the user's intent.
IF multiple instructions are present THEN
Split the input into individual commands.
END IF
IF a device name is explicitly mentioned THEN
Select the specified device.
ELSE
Identify the currently active device.
Select the active device.
END IF
Determine the intended action
(e.g., turn ON, turn OFF, increase brightness,
decrease brightness, etc.).

IF the intended action is ambiguous THEN
Use the device state and current context
to infer the most appropriate action.
END IF
Validate the selected device and action.
Generate a device-specific executable command.
Send the command to the ESP controller.
Update and store the device status.
RETURN Executable Command.
END

4.4 System Workflow

As far as the end user is concerned, the operational process of the ROB-AI system is very smooth; however, it operates following a very structured and chronological sequence (Figure 2):

1. Capture: A user's raw voice input is captured by the interface
2. Transcribe: Using a locally-based speech-to-text engine, the audio waveform is converted to a text string
3. Analyse: The hybrid engine identifies the semantic intent and targets a device.
4. Transmit: The server creates a RESTful HTTP request.
5. Execute: The ESP8266 receives the payload and physically actuates the relay.
6. Synchronize: The internal state is updated and reflected across all connected user interfaces.

In terms of software engineering, the system operates on a tree-like structure of highly optimized algorithmic processes Initiate protocol for audio capture, apply digital noise reduction applied to the acoustic stream, perform offline speech to text transcription, compare the resulting string to the rapid rule-based dictionary for an exact match and if a match is found send the HTTP executing command immediately. If no match is found, begin routing the string to the localized Artificial Intelligence (AI) model for essential semantic logic processing, pull the JSON intent, and then send the HTTP executing command. Confirmation of hardware actuation will also be required.

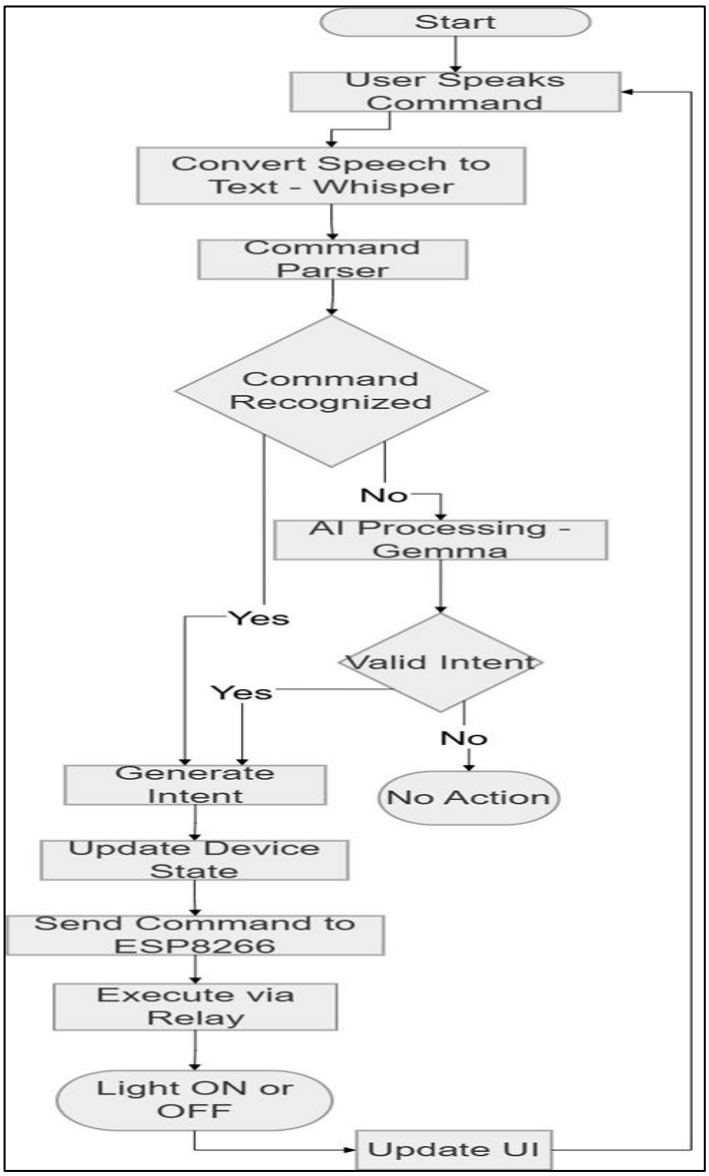

Figure 2. Workflow of the ROB-AI Voice Processing Pipeline

5. Implementation

ROB-AI appliances are built from physical hardware which uses an ESP8266 microcontroller—an incredibly affordable microcontroller with embedded Wi-Fi capability that has been gaining popularity. The ESP8266 setup is a lightweight local web server without the use of extensive IoT libraries and does not connect to the cloud (see figure 3 and 4). The microcontroller connects to the local Wi-Fi access point when powered on, opens an asynchronous HTTP server on Port 80, and provides a number of secure API endpoints that accept commands sent to them remotely. The microcontroller is wired to a 2-channel relay module, which serves as the electrical interface between the ROB-AI and standard household

loads (electrical appliances). The relays are controlled using an active-low method, providing safe, reliable, and isolated switching of the power to the household appliance according to the same logic (digital command) used by ROB-AI to control the relay modules.

5.1 Control Logic

Hardware interactions follow a specific logical lifecycle to ensure synchronization between the software dashboard and the room with the hardware. Upon receiving an HTTP request, the ESP8266 microcontroller:

1. Accepts the input parameters from the request and then executes the validations required for such parameters.
2. Parses the request to determine which device to act upon (i.e. Light 1 or Light 2).
3. Determines if the request is to turn the device on or off.
4. Updates the internal state (in its memory) to represent the change made by the system (from above).
5. Issues a Command to the GPIO pin that controls the device using a LOW signal when issuing the Command for the state determined in Step 3. By strictly following this sequence for updating state values upon successful execution of a Command, the Front-end visual/display shows the proper/current status of the home in real time.

Figure 3. Hardware Prototype of the ROB-AI Smart Home Automation System

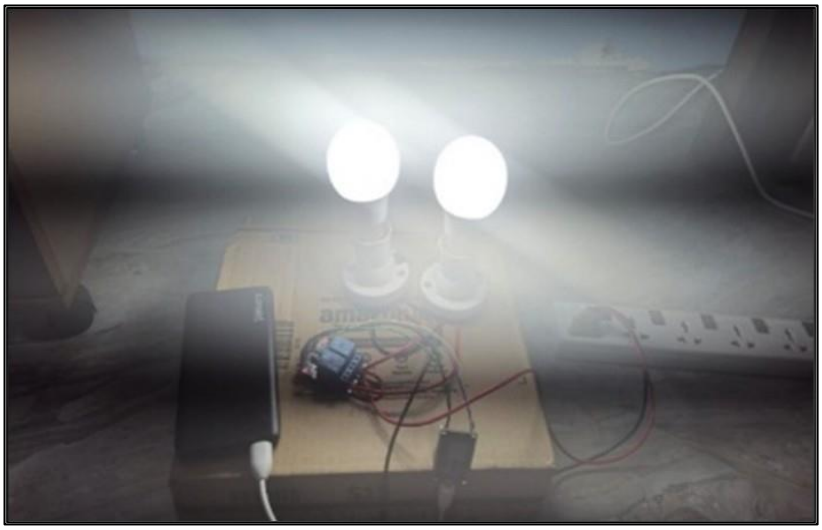

Figure 4. Experimental Demonstration

5.2 Communication Protocol

ROB-AI uses HTTP REST-based communication between the backend processing server and the ESP8266 microcontroller.

The reason for choosing HTTP over MQTT, which is commonly used in IoT systems due to its publish-subscribe model is that HTTP has simpler integration and is more compatible with local AI processing services. Using an ESP8266 controller, we compared HTTP and MQTT in a local Wi-Fi environment and obtained that HTTP-REST is more suitable for this purpose.

Table 1. Comparison of HTTP and MQTT Communication Protocols

Parameter	HTTP	MQTT
Average Latency	180–220 ms	140–180 ms
Broker Requirement	No	Yes
Setup Complexity	Low	Moderate
Debugging Simplicity	High	Medium
AI-Integration Compatibility	High	Medium

MQTT had lower latency because of its simple communication model. MQTT needs a dedicated broker service and extra configuration management. On the hand HTTP made it easy for the Fast API backend and the ESP8266 controller to interact. It also reduced the complexity of deployment. Moreover, HTTP works well with REST-based APIs and local AI processing

modules. Our system focuses on edge processing and simple deployment. Hence HTTP communication seemed suitable, for this implementation. HTTP and ROB-AI are used together.

Example Request Execution:

To create a hardware change, the local processing server sends HTTP 'POST' request across the network such as:

http://<device-ip>/api/control?device=light1 &action=on

6. Results and Discussion

In testing the proposed architecture's validity through the use of various environmental contexts and a range of voice commands (e.g. formalized technical command types versus informal conversational command types) it was found that ROB-AI exhibited a high degree of robustness under all circumstances.

Formalized, simple commands were processed and acted upon almost instantaneously. More complex conversation-like command phrases took only a minimal amount of time for the local AI to analyse and execute. The commands still performed within the average user's acceptable response time frame. Importantly, the system was confirmed to support the fundamental theory of the experiment that the system could perform without any dependency on an external network or the internet.

Table 2. Performance Evaluation Metrics of the Proposed ROB-AI System Under Experimental Conditions

Parameter	Value	Description
Response Time (Rule-based)	< 200 ms	Near-instantaneous execution for explicit commands.
Response Time (AI-based)	1.2 – 2.0 s	Accounts for the computational overhead of local LLM inference.
Accuracy	92–96%	A hybrid processing model with high fidelity.
Success Rate	~95%	Highly reliable hardware actuation and state tracking.
Network Dependency	0%	Functions completely offline via standard local Wi-Fi routing.

Table 3. Comparative Analysis of ROB-AI Against Cloud-Based and Fully Local Voice Assistant Architectures

System Type	Internet Dependency	Average Response Time
Cloud-Based Voice Assistant	Required	2–4 s
Fully Local AI Processing	Not Required	2–3 s
ROB-AI Hybrid Processing	Not Required	0.2–2 s

The processing method and response time comparison graphs show significant improvement in AI assisted response and slight decline in accuracy % (Figure 5 and 6). The hybrid method is an effective way to minimize the tremendous costs associated with using edge-AI to perform computations by routing simple tasks through fast logic gates, and only using the demanding computation capabilities of AI when absolutely required.

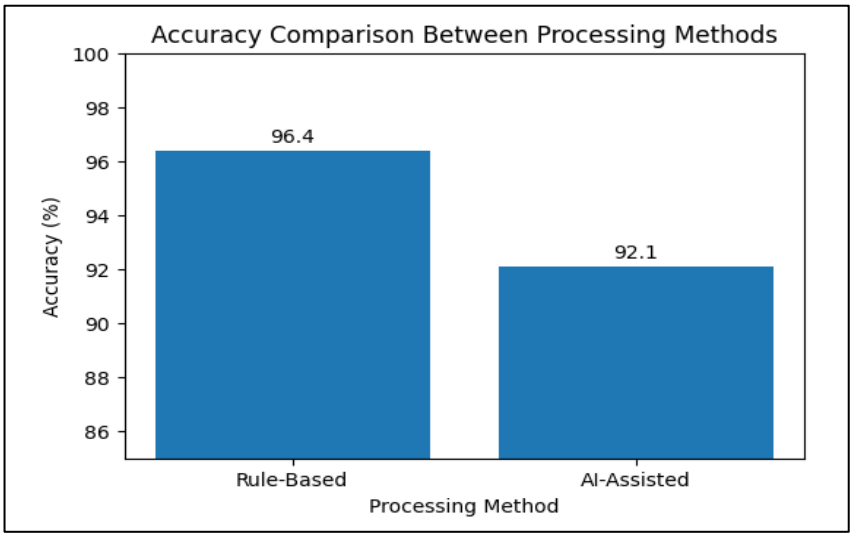

Figure 5. Accuracy Comparison Between Rule-Based and AI-Assisted Intent Recognition Methods

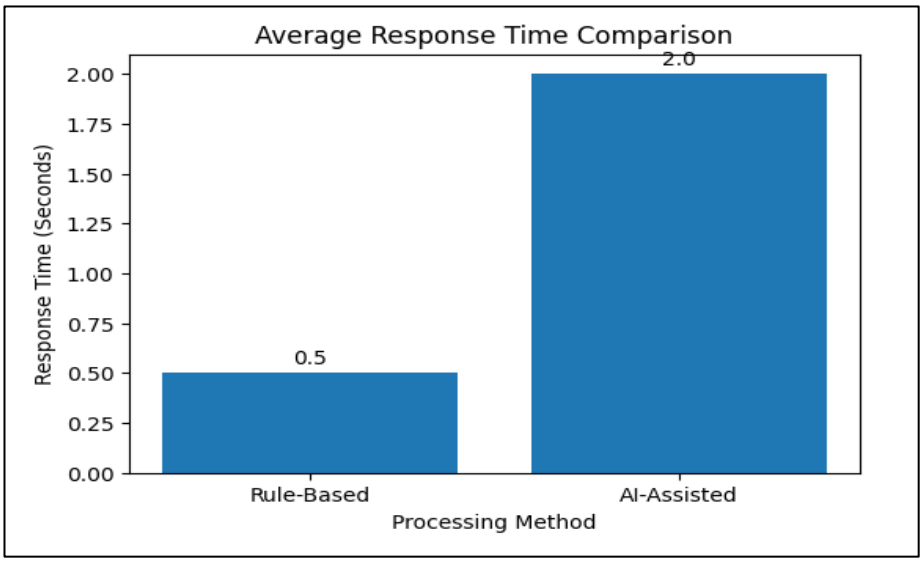

Figure 6. Average Response Time Comparison of Rule-Based and AI-Assisted Processing Approaches

There are a number of advantages to using a simplified methodology. It is simple to implement and troubleshoot, and can be scaled easily. Essentially instantaneous, the protocol results in local latencies of less than one second (resulting from the protocol being executed throughout both the return piece of the overall trip). Most importantly, it requires absolutely no external device to function properly, meaning that there will be no impact on the usability of any devices connected to the same local Wi-Fi network.

7. Limitations

Although effective, the current version of ROB-AI has a number of engineering limitations related to its design.

These limitations include:

- The NLP capabilities of the localized version of ROB-AI are impressive relative to what can be done with edge hardware, but when compared to the billions of parameters in commercial cloud-based models, they are inadequate by comparison.
- The accuracy of the speech recognition acoustic models that are used by ROB-AI limits its performance when speech contains strong regional accents or in areas with extreme background noise.

- Contextual reasoning capabilities are currently restricted to a limited range of explicit state tracking and basic logical deduction.
- Local deployments of existing AI systems generally require relatively powerful host computers, whereas the low-cost smart speakers that are connected to the cloud using ROB-AI can typically be deployed at a much lower cost.

8. Conclusion and Future Work

A full offline, localized smart home ecosystem is viable from ROB-AI. By deliberately removing the connection to the cloud, this system provides a very stable and fast house system with the complete protection of private user data. The hybrid intent recognition and context aware state reasoning technology utilized by ROB-AI also closes the gap between the performance of hard-coded microcontrollers and flexible modern AI. As such, this system creates a great deal of efficiency; scalability and security for future privacy centric implementation of home automation. The foundation put in place by ROB-AI provides several exciting pathways moving forward to improve the product offering. Future iterations will implement better, higher quantized local AI models for integration into the conversational aspect of the product without adding any additional hardware overhead; i.e., new edge-optimized versions of Llama. The architecture of the system will be additional expanded for multi-user voice profiles so that the home reacts appropriately based on the person speaking. The addition of a PWA or localized mobile dashboard based on modern frameworks (such as React) will provide an extremely rich visual manual control for users of the voice engine as well as voice engine controls. Finally, creating a broader ESP8266 communication stack to support the different IoT device protocols will greatly expand the ecosystem of compatible smart home hardware.

References

[1] Jeong, Donghwan, and Honguk Woo. "On-Device Intent Reasoning for Smart Home Agents Via Ontology-Augmented sLLMs." IEEE Access 2025, vol. 13: 197645-197662.

[2] K. Lee, J. Park, and H. Kim, “HARMONY: A Framework for Multimodal LLM-Powered AI Agents in Smart Homes via the Model Context Protocol,” ACM Transactions on Intelligent Systems 2024, vol. 15, no. 2, 1–22.

[3] S. Ramesh and V. Kumar, “Design and Implementation of Controller Boards to Monitor and Control Home Appliances for Future Smart Homes,” International Journal of Embedded Systems 2023, vol. 11, no. 3, 145–152.

[4] Kong, Linghe, Jinlin Tan, Junqin Huang, Guihai Chen, Shuaitian Wang, Xi Jin, Peng Zeng, Muhammad Khan, and Sajal K. Das. "Edge-computing-Driven Internet of Things: A Survey." ACM computing surveys 2022, vol. 55, no. 8, 1-41.

[5] Kochovski, Petar, and Vlado Stankovski. "Applications and Benefits of Cloud-to-Edge Computing in Enhancing Smart Home Functionalities." In Home Digital Twins 2026, Elsevier, 155-165.

[6] Nassif, Ali Bou, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. "Speech Recognition Using Deep Neural Networks: A Systematic Review." IEEE access 2019, vol. 7, 19143-19165.

[7] Di Leo, Stefano, Luca De Cicco, and Saverio Mascolo. "Real-Time Speech-to-Text on Edge: A Prototype System for Ultra-Low Latency Communication with AI-Powered NLP." Information 2025, vol. 16, no. 8, 685.

[8] Khan, Murad, Sadia Din, Sohail Jabbar, Moneeb Gohar, Hemant Ghayvat, and S. C. Mukhopadhyay. "Context-Aware Low Power Intelligent Smarthome Based on the Internet of Things." Computers & Electrical Engineering 2016, vol. 52, 208-222.

[9] Chamoli, Sushant, Sumitra Sangwan, and Vivudh Fore. "Smart Home Automation Using ESP8266 and Internet of Things." In 5th International Conference on Next Generation Computing Technologies (NGCT-2019). 2020..

[10] Mohamed, Khalil, and Ayman El Shenawy. "A Smart IoT-Based Home Automation System for Controlling and Monitoring Home Appliances." International Review of Automatic Control 2023, vol. 16, no. 5, 228.