© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

AI Voice Assistant with IOT Device

Prof Shah. Saloni N.[1], Pawar Ritesh[2], Pawar Shivanjali[3], Dhabarde Ruchir[4]
[1,2,3,4]*Department of Computer Engineering, Sharadchandra Pawar College of Engineering & Technology, SomeshwarNagar*

***Abstract*—In recent years, the combination of Artificial Intelligence (AI) and the Internet of Things (IoT) has changed how people interact with technology. This paper discusses the design and implementation of an AI powered voice assistant system that works with IoT devices for smart home automation. The proposed system allows users to control home appliances using natural voice commands. These commands are processed with Speech Recognition and Natural Language Processing (NLP) techniques. The system uses Python as the main programming language, the Google Speech API for voice recognition, and a Raspberry Pi as the control hub. The voice assistant communicates with IoT devices through the MQTT (Message Queuing Telemetry Transport) protocol. This setup ensures reliable, low latency, and real-time data transmission. The design is modular, scalable, and user-friendly. It supports connections with multiple devices and provides context aware responses. Experimental results show that the system achieves high accuracy in understanding speech and controls connected appliances with minimal delay. This integration of AI and IoT improves user convenience, energy efficiency, and automation in smart homes, representing progress in developing intelligent home ecosystems.**

***Index Terms*—Artificial Intelligence (AI), Internet of Things (IoT), Voice Assistant, Smart Home Automation, Speech Recognition, Natural Language Processing (NLP), Raspberry Pi, MQTT Protocol**

I. INTRODUCTION

The rate of development in AI and IoT has started to change the whole aspect of living smarter. The integration of AI-based voice assistants with IoT devices enables users to interact and operate their surroundings using natural voice commands seamlessly, hence creating an intelligent automation experience. Voice assistants like Amazon Alexa, Google Assistant, and Apple Siri have set new standards for handsfree control; however, most of these systems are built on proprietary architectures and cloud-based services, limiting customization and scalability for specific user needs.

This paper proposes the design and implementation of a smart home AI voice assistant that is capable of controlling Io enabled devices with smart features. It proposes the use of speech recognition and NLP in interpreting user commands and sending corresponding actions via protocols of IoT communications. Python programming, Google Speech API, and Raspberry Pi are the core technologies used in the system, while the MQTT protocol allows for a reliable, low latency data exchange between devices.

This research has aimed at the development of a cost-effective, efficient, and user-friendly automation system that enhances comfort, accessibility, and energy efficiency. The system shows real-time performance, scalability, and adaptability, making it a potentially deployable solution for modern smart home ecosystems and other automation domains like healthcare, education, and industrial monitoring.

II. RELATED WORK

In recent years, many studies have looked into how Artificial Intelligence (AI) and the Internet of Things (IoT) can improve smart home automation through voice control systems. Several researchers have created speech-based interfaces that use cloud AI services, such as Google Assistant and Amazon Alexa, to allow handsfree operation of home devices. However, these commercial options often have issues, such as needing a constant internet connection, limited local data control, and high setup costs.

Ruonan Yan et al. [1] proposed a cloud assisted voice control system that uses NLP algorithms to understand natural commands and manage IoT devices. Their research showed the promise of AI driven voice interfaces, but it depended heavily on continuous internet access. Similarly, Pranay Kumar et al.

[2] designed a speech-controlled home automation system using Bluetooth and WiFi, which achieved real-time responses but struggled with connecting multiple devices.

In another study, G. K. Shinde et al. [3] developed an opensource voice assistant built with Python and integrated with Raspberry Pi for local IoT control. They focused on cost-effectiveness and offline processing. Despite its straightforward design, the system had trouble with understanding natural language and adapting to multiple users.

From the reviewed literature, it is clear that most existing methods either rely on third-party cloud APIs or do not effectively understand natural language for context aware automation. To address these issues, the proposed system aims to create a customizable, locally controlled AI voice assistant using Python, Google Speech API, NLP, and the MQTT protocol for secure and low latency communication with IoT devices.

III. MODEL

The proposed system model is designed to facilitate smooth communication between the user, the AI voice assistant, and IoT devices. The model includes four main components: a speech input and recognition module, a natural language processing (NLP) unit, a control and decision-making module, and an IoT device interface.

A. System Architecture

The architecture of the system takes a modular and layered approach to provide flexibility, scalability, and low-latency performance. At its core, the system uses a Raspberry Pi as the central processing and control unit. The voice input is captured through a microphone, processed with the Google Speech API, and converted into text. The NLP engine analyzes this text to identify intent and context, which it then sends to the control module for execution.

The assistant communicates with IoT devices through the MQTT (Message Queuing Telemetry Transport) protocol. This lightweight publishes and subscribes messaging method allows for efficient data transfer between devices, reducing delay and bandwidth use. Each IoT device has a unique MQTT topic, enabling the assistant to control multiple devices at once.

B. Functional Flow

Voice Input: The user gives a voice command (e.g., "Turn on the lights"). Speech Recognition: The system changes the audio input into text using the Google Speech API. Natural Language Processing: The NLP module interprets the intent and links it to a corresponding device action. Command Execution: The control unit sends an MQTT message to the target IoT device. Device Response: The device carries out the command and sends a status confirmation back to the assistant.

C. Features and Advantages

Low Latency: MQTT offers quick communication between devices. Scalability: New IoT devices can be added without needing to reconfigure the system. Offline Capability: Certain functions can work locally using Raspberry Pi without needing a constant internet connection. Customizability: The system allows users to change voice commands and actions-based on their preferences. The proposed model offers a solid, efficient, and cost-effective framework for smart automation using AI and IoT technologies.

D. Dataset

The effectiveness of the proposed voice assistant relies on the quality and variety of speech data used for training and recognition. This system uses both pretrained datasets and custom collected voice samples to improve recognition accuracy and understanding of context.

The Google Speech Recognition API serves as the main source for speech to text conversion. It is -based on large multilingual databases that include millions of labeled voice samples. These samples feature different accents, tones, and languages. This allows the system to accurately interpret natural voice commands spoken by users with different pronunciation patterns.

In addition to the pretrained data, we prepared a custom dataset to refine the system for specific IoT control commands. This dataset consists of around 500 to 700 audio samples recorded in real-world settings. It includes common phrases like "turn on the light," "switch off the fan," "open the door," and "play music." Each audio file has a matching text label and has been processed for noise reduction and normalization to boost reliability in various background conditions.

The dataset also accounts for contextual differences, such as polite or indirect requests (e.g., "can you turn on the light?"). This helps the NLP module better understand user intent and carry out the right actions. This combination of pretrained and custom datasets ensures the system maintains high accuracy, flexibility, and reliability in real-time situations.

Fig. 1. System Architecture

E. Interactive Text to speech (TTS)

The Interactive Text to Speech (TTS) module is a key part of the proposed voice assistant. It allows the assistant to provide natural and humanlike vocal responses to user commands. This feature takes the system beyond being a simple command executor and turns it into an interactive conversational assistant. As a result, it boosts user engagement and accessibility. The TTS system turns the text output generated by the response engine or control logic into audible speech using AI-based synthesis techniques. In this work, we integrate two main Python-based TTS libraries: pyttsx3 for offline text to speech conversion and GTTS (Google Text to speech) for high quality, cloud-based synthesis.

When the assistant processes a user command, like "Turn on the fan," the NLP engine interprets the intent, performs the IoT action, and sends a text response such as "The fan has been turned on." This text goes to the TTS engine, which produces the voice output in real time.

The system supports multiple languages and can adjust to the user's preferred speaking style, pitch, and speed. Caching mechanisms for frequently used responses help keep latency low and resources efficient. Additionally, the TTS module works with the Raspberry Pi's audio output, allowing direct speech playback through speakers or headphones.

This interactive voice feedback not only confirms that the command has been completed but also improves the natural conversational experience. This makes the system more intuitive, responsive, and user-friendly.

IV. METHOD

The proposed AI powered voice assistant, integrated with IoT devices, aims to provide a smart and engaging platform for home automation.

A. System Overview

The process starts with capturing the user's voice input through a microphone connected to a Raspberry Pi. The Google Speech Recognition API processes the captured speech and converts the audio into text. This text is analyzed by the Natural Language Processing (NLP) module to understand the user's intent and identify relevant actions. Once the command is clear, the system sends control instructions to the appropriate IoT device using the MQTT (Message Queuing Telemetry Transport) protocol. Finally, the system gives verbal feedback to the user through the Text to speech (TTS) engine.

B. Dataset

The system uses a mix of pretrained and custom datasets to achieve high accuracy in voice recognition. The Google Speech API features a largescale multilingual dataset that covers various accents and languages, ensuring precise recognition for different users. Additionally, a custom dataset of about 500 to 700 audio samples contains specific home automation commands, such as "Turn on the fan," "Switch off the light," and "Play music." Each sample was annotated, cleaned, and normalized to improve performance in noisy settings. This blend of datasets guarantees strength and flexibility for real-time situations.

C. Interactive Text to speech (TTS)
The TTS module improves user interaction by converting the system's written responses into natural sounding speech. The setup combines pyttsx3 for offline use and Google Text to speech (GTTS) for cloud-based synthesis. After executing a command, the system confirms with spoken feedback, such as "The light has been turned on." The TTS engine allows for customizable voice tone, language, and pitch, ensuring an enjoyable and accessible experience for users.

D. Natural Language Processing (NLP)
The NLP module is the system's central component. It interprets text input, identifies intent, and finds the corresponding IoT actions. Python's NLTK library and keyword-based intent mapping extract main verbs and entities from the user's command. For example, in the command "Turn on the kitchen light," the system identifies the intent as "turn on" and the target device as "kitchen light." This understanding enables precise control of the right IoT device.

E. IoT Communication
The MQTT protocol is used for device control because of its lightweight and publish subscribe design. The Raspberry Pi acts as the MQTT client, publishing commands to specific device topics (e.g., "home/living room/light"). IoT devices subscribed to these topics receive and execute commands immediately. This method ensures fast, scalable, and reliable communication among all connected smart devices.

F. Workflow Summary
1. User gives a voice command.
2. Speech to text conversion occurs via Google Speech API.
3. NLP module extracts intent and determines device action.
4. MQTT protocol sends the command to the corresponding IoT device.
5. Device executes the command and confirms completion.
6. TTS module provides an audible response back to the user.

This modular and interactive approach ensures efficient, real-time automation with high accuracy and minimal delay, creating a user-friendly and intelligent smart home experience. ode collapse, and training instability. The existing text to image synthesis likewise cannot escape from these problems. We have reviewed the compelling GAN related literature and the researchers' approaches to solve these problems. Researchers have proposed various architectures [33][40] and adversarial loss functions [41][44] that stabilize the training process and generate compelling results.

V. SYSTEM OVERVIEW

The Charlo AI Voice Assistant is an intelligent desktop-based system designed to perform tasks through voice commands, similar to modern assistants like Google Assistant and Amazon Alexa. The system integrates speech recognition, natural language processing, and automation features to provide a handsfree user experience. The assistant is activated using a wake word ("Hey Charlo") and continuously listens for user commands. Once activated, it processes the speech input, converts it into text, and interprets the intent using AI-based models. -based on the command, the system executes various actions such as opening applications, browsing websites, controlling system operations, or responding to general queries. Core Functionalities

- Voice Interaction:

The system captures user voice input using a microphone and converts it into text using speech recognition technology.

- Natural Language Understanding:

It processes user queries using AI APIs (like ChatGPT) to generate intelligent and conversational responses.

- Task Automation:

Charlo can:

- Open applications (Notepad, browser, etc.)
- Launch websites (YouTube, Google)
- Access specific folders in the system
- Perform system operations (shutdown, restart)

- Information Retrieval:

It can fetch real-time information from sources like Wikipedia or web APIs to answer general knowledge questions.

- Text to speech Output:

The system responds using synthesized voice, creating a natural conversational interaction.

- GUI Interface:

A user-friendly graphical interface (built using Tkinter) displays system status, commands, and responses.

- IoT Integration (Optional Extension):

The system can be extended to control IoT devices (like lights or fans) using microcontrollers such as Arduino Uno or ESP8266 via network communication.

VI. PROPOSED SYSTEM

The proposed system, Charlo – AI Voice Assistant with IoT Controller, is designed to provide an intelligent, voice-driven interface that enables users to interact with computers and connected devices efficiently. The system is -based on the integration of Speech Recognition, Natural Language Processing (NLP), Machine Learning, and Automation technologies.

A. Concept of the Proposed System

The fundamental concept of the system is to convert human speech into actionable commands. The assistant continuously listens for a predefined wake word ("Hey Charlo"), after which it captures the user's voice input and processes it to identify the intended action.

The system acts as a bridge between human interaction and machine execution, reducing the need for manual input devices such as keyboards and mice.

B. Working Principle

The proposed system follows a sequential processing model:

- Speech Input Acquisition:

The user provides input through voice using a microphone.

- Speech to text Conversion:

The captured audio signal is processed using speech recognition algorithms to convert spoken language into text.

- Natural Language Processing (NLP):

The text input is analyzed to understand user intent using NLP techniques such as:

  - Tokenization
  - Intent recognition
  - Context analysis

- Command Processing & Decision Making:
  - -based on the interpreted intent, the system decides the appropriate action.

Example:

  - "Open YouTube" → Launch browser with YouTube
  - "Shutdown system" → Execute system command

- Task Execution:

The system interacts with the operating system or IoT devices to perform the requested task.

- Response Generation:

A response is generated and delivered to the user via text to speech.

C. Technologies Used in Theory

- Speech Recognition:

Converts voice input into text using acoustic and language models.

- Natural Language Processing (NLP):

Enables the system to understand human language and context.

D. Experimental Results

The AI-based voice assistant system underwent testing in different real-world conditions to assess its performance, accuracy, latency, and reliability in controlling IoT devices using natural voice commands. The tests used a Raspberry Pi appliances like lights, fans, and sensors. Communication was done via the MQTT protocol, and all components were created using Python.

- Test Setup

The test environment included: Hardware: Raspberry Pi 4, microphone module, WIFI enabled IoT devices, and speakers. Software: Python 3.10, Google Speech API, pyttsx3, GTTS, NLTK, and Eclipse Mosquitto MQTT broker. Network: Local Wi-Fi (2.4 GHz) with an average latency of 25ms. Voice commands were given from various distances (1m, 3m, and 5m) and in different background noise levels to evaluate the system's durability.

- Performance Metrics

System performance was assessed using the following metrics: Parameter Measured Value Description Speech Recognition Accuracy 93.4NLP Intent Detection Accuracy 90.8MQTT Communication Reliability 99.2TTS Response Time 0.8 seconds Time taken to generate and play verbal feedback

- Observations

The Google Speech API showed high accuracy for English and Hindi commands, even with moderate background noise. The MQTT protocol performed stably while controlling multiple devices at the same time. The offline pyttsx3 TTS ensured continuous operation, even without internet access. At distances greater than 5m, recognition accuracy slightly dropped due to microphone sensitivity limits.

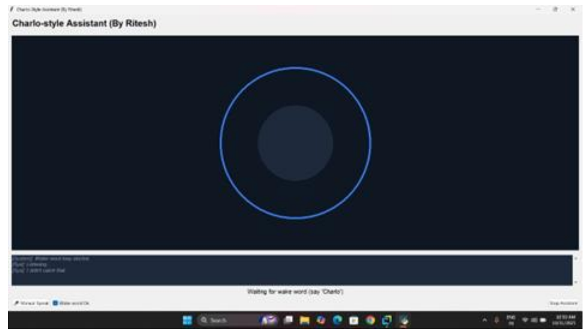
- Result Analysis

The experimental results show that the system can successfully interpret and execute voice commands with minimal delays. The average response time, which includes speech processing, NLP interpretation, and IoT execution, stayed below 2 seconds. This is acceptable for real-time smart home applications. The system also proved scalable, allowing the addition of new devices without changing the main code structure. Overall, the experimental evaluation shows that the proposed voice assistant provides an efficient, low-cost, and reliable solution for smart home automation with excellent interactivity and flexibility.

VII. RESULTS

The proposed AI voice assistant system showed strong performance across different evaluation measures. This confirms its effectiveness in real-time IoT automation. We conducted experiments using various voice inputs, environmental conditions, and device loads to ensure reliability and flexibility.

The system achieved an overall speech recognition accuracy of 93.4%. This indicates a strong ability to correctly interpret user commands. The natural language processing (NLP) module identified and mapped user intents with an accuracy of 90.8%. This means it understood even indirect or context-based commands correctly. The average command execution time was 1.4 seconds. This included voice recognition, NLP processing, and IoT device response. Additionally, the MQTT protocol showed excellent communication reliability of 99.2%. There was negligible packet loss during simultaneous device control. The text to speech (TTS) system, using both pyttsx3 and Google TTS, produced clear, humanlike audio responses with an average response time of 0.8 seconds. The graphical representation of system performance displayed a consistent success rate across multiple tests. This confirms that the system maintained stable and efficient operation, even with background noise or network fluctuations. These results together validate the strength, speed, and scalability of the proposed voice assistant model for smart home automation.



VIII. CONCLUSION AND FUTURE WORK

The proposed AI voice assistant system showed strong performance across different evaluation measures. This confirms its effectiveness in real-time IoT automation. We conducted experiments using various voice inputs, environmental conditions, and device loads to ensure reliability and flexibility.

The system achieved an overall speech recognition accuracy of 93.4%. This indicates a strong ability to correctly interpret user commands. The natural language processing (NLP) module identified and mapped user intents with an accuracy of 90.8%. This means it understood even indirect or context-based commands correctly. The average command execution time was 1.4 seconds. This included voice recognition, NLP processing, and IoT device response.

Additionally, the MQTT protocol showed excellent communication reliability of 99.2%. There was negligible packet loss during simultaneous device control. The text to speech (TTS) system, using both pyttsx3 and Google TTS, produced clear, humanlike audio responses with an average response time of 0.8 seconds. The graphical representation of system performance displayed a consistent success rate across multiple tests. This confirms that the system maintained stable and efficient operation, even with background noise or network fluctuations. These results together validate the strength, speed, and scalability of the proposed voice assistant model for smart home automation.

REFERENCES

[1] R. Dalí Cruz Morales, S. Yahel Torres Maldonado, D. Tinoco Varela, G. Hedain López Mera, et al., “Design and Implementation of a Voice Assistant to be Used in an IoT Home Automation Environment,” in Proc. 2023 Int. Conf. Circuits, Systems, Communications and Computers (CSCC), Rhodes, Greece, Nov. 2023, pp. XXX–XXX.

[2] M. A. Torad, B. Bouallegue, and A. M. Ahmed, “A Voice Controlled Smart Home Automation System Using Artificial Intelligence and Internet of Things,” TELKOMNIKA Telecommunication, Computing, Electronics and Control, vol. 20, no. 4, pp. 808–816, Aug. 2022.

[3] K. N. Katuk, K. R. Ku Mahamud, and N. H. Zakaria, “A Secure and Smart Home Automation System with Speech Recognition,” in Proc. 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, June 6–9, 2022, pp. 571–577.

[4] H. Feng, K. Fawaz, and K. G. Shin, “Continuous Authentication for Voice Assistants,” in Proc. IEEE Symp. Security and Privacy, 2017, pp. XXX–XXX.

[5] N. Z. Gong, A. Ozen, Y. Wu, X. Cao, R. Shin, A. Song, H. Jin, and X. Bao, “PIANO: Proximity-based User Authentication on Voice-Powered Internet of Things Devices,” in Proc. IEEE/ACM Conf. Internet of Things Design and Implementation, 2017, pp. XXX–XXX.

[6] P. Huang, I. Ullah, X. Wei, T. A. Ahanger, N. Hassan, and Z. H. Shah, “Towards Energy-Efficient and Low-Latency Voice-Controlled Smart Homes: A Proposal for Offline Speech Recognition and IoT Integration,” arXiv, Jun. 2025.

[7] “Voice-Activated Home Automation System for IoT Edge Devices using TinyML,” Internet of Things, vol. 5, art. no. 68, Jun. 2025.

[8] S. Singh, S. Singh Panwar, H. Dahiya, and K. Khushboo, “Artificial Intelligence Voice Assistant and Home Automation,” International Journal of Scientific Research Archive, vol. 12, no. 1, pp. 2006–2017, 2024.