IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

TARS-D AI Voice Assistant

Prof. Vedasree T K[1], Aditya[2], Ankith S B[3], Ayush H M[4], K Srastick S[5]

Assistant Professor Dept. of ISE, Bangalore Institute of Technology Bangalore, India[1]

Student, Dept. of ISE, Bangalore Institute of Technology Bangalore, India[2-5]

Abstract: The rapid advancement of artificial intelligence has significantly transformed human–computer interaction, enabling the development of intelligent virtual assistants capable of understanding and responding to natural language. This paper presents the design and implementation of TARS-D, a desktop-based AI voice assistant that facilitates seamless interaction through both voice and text-based commands. The system integrates speech recognition, natural language processing (NLP), and machine learning techniques to interpret user intent and execute a wide range of system-level operations. It utilizes speech-to-text conversion for capturing user input and text-to-speech synthesis for generating natural responses. The assistant is capable of performing tasks such as file and folder management, application control, web browsing, scheduling, and information retrieval. A key feature of TARS-D is its emphasis on privacy and offline functionality, as it processes user data locally rather than relying heavily on cloud services. The modular architecture of the system ensures scalability and ease of integration of new features. Additionally, the assistant improves accessibility by enabling hands-free interaction, making it beneficial for users with visual or physical impairments. The proposed system demonstrates an efficient, secure, and user-friendly solution for desktop automation, contributing to enhanced productivity and improved user experience in modern computing environments.

Keywords: Artificial Intelligence, Voice Assistant, Natural Language Processing, Speech Recognition, Desktop Automation, Human–Computer Interaction.

I. INTRODUCTION

The evolution of artificial intelligence (AI) has significantly enhanced the way humans interact with computing systems, shifting from traditional input methods such as keyboards and mice to more natural and intuitive interfaces. Among these advancements, voice-based interaction has emerged as a powerful paradigm, enabling users to communicate with systems using natural language. AI-powered voice assistants have gained widespread adoption due to their ability to simplify tasks, improve accessibility, and enhance user experience.

Despite the availability of popular voice assistants, most existing solutions are heavily dependent on cloud infrastructure, raising concerns related to data privacy, latency, and reliability in environments with limited or no internet connectivity. Furthermore, many of these systems offer limited customization and lack deep integration with desktop-level operations, restricting their usability for advanced and personalized tasks. To address these challenges, this paper presents TARS-D, an intelligent desktop-based AI voice assistant designed to enable seamless human–computer interaction through voice and text commands. The system integrates speech recognition, natural language processing (NLP), and text-to-speech technologies to understand user input, interpret intent, and generate appropriate responses. Unlike conventional assistants, TARS-D emphasizes local processing, ensuring enhanced privacy, reduced dependency on internet connectivity, and improved system responsiveness. The assistant is capable of executing a wide range of tasks, including application control, file management, web browsing, scheduling, and information retrieval. Its modular architecture allows easy scalability and integration of additional features, making it adaptable to evolving user requirements. Moreover, the system promotes accessibility by enabling hands-free interaction, which is particularly beneficial for users with visual impairments or mobility limitations. Overall, TARS-D aims to provide a secure, efficient, and user-friendly desktop automation solution that bridges the gap between advanced AI capabilities and practical everyday computing needs.

II. LITERATURE REVIEW

A. Voice Assistant Using Python and AI

- This work integrates Python with AI and NLP techniques to develop a multifunctional voice assistant capable of performing tasks such as web searches, messaging, and application control.
- Although the system offers a wide range of functionalities, it requires significant computational resources and depends on internet connectivity for speech processing.

B. NLP Based AI Voice Assistant

- This research proposes a structured three-phase model consisting of speech recognition, intent mapping,

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

and response generation.

- The system improves efficiency through organized workflow design; however, it relies on predefined mappings, which restrict scalability and adaptability to varied user inputs.

C. AI Based Virtual Assistant

- This assistant leverages Python and speech processing libraries to simulate natural human interaction.
- It enhances usability and supports multiple desktop operations but lacks advanced contextual understanding and adaptive learning capabilities, making it less effective for complex interactions.

D. NLP Based AI Voice Assistant

- The system focuses on improving accessibility through voice-driven control of desktop operations.
- It uses NLP techniques to interpret commands and execute tasks efficiently.

E. JARVIS: An AI Voice Assistant

- This desktop-based assistant is capable of performing tasks such as sending emails, reading documents, and controlling applications.
- It emphasizes offline functionality and user privacy but lacks multi-turn conversation handling and deeper contextual understanding

F. Voice Assistant Using Artificial Intelligence

- This system integrates multiple Python libraries to perform communication and multimedia tasks.
- It provides a feature-rich environment but relies on external APIs for speech recognition, making it less suitable for offline usage and raising concerns about reliability.

G. Desktop Voice Assistant for Visually Impaired

- This work focuses on accessibility by enabling visually impaired users to interact with computers using voice commands.
- While it improves usability and independence, it depends on internet-based APIs and lacks advanced NLP-based intent recognition.

H. Desktop Voice Guide Using Python and AI

- This system follows a modular architecture with separate components for speech recognition, processing, and response generation.
- Although the modular design improves maintainability, the reliance on keyword matching limits flexibility and natural language understanding.

I. Python-Based Desktop Assistant with Voice Recognition

- This assistant performs basic desktop tasks using speech recognition and text-to-speech technologies.
- It is simple and efficient but lacks advanced AI capabilities such as contextual awareness, learning ability, and intelligent decision-making.

J. Summary of Literature

- From the reviewed literature, it is evident that existing systems primarily rely on predefined commands, lack contextual understanding, and often depend on cloud-based services. These limitations highlight the need for a more intelligent, scalable, and privacy-focused solution.
- The proposed addresses these challenges by incorporating natural language processing, modular architecture, and local processing capabilities to deliver a more adaptive, secure, and efficient desktop assistant.

III. TOOLS AND TECHNOLOGIES

A. Architecture

The architecture of the TARS-D AI Voice Assistant is designed using a modular software framework that enables efficient voice-based interaction and task execution. The system consists of the following components:

1. Input Module
   - Captures user input through voice or text
   - Uses a microphone for real-time speech acquisition
2. Processing Module
   - Converts speech into text using speech recognition
   - Applies Natural Language Processing (NLP) to interpret commands
3. Control Module
   - Maps user intent to predefined system functions
   - Executes commands using backend logic
4. Output Module
   - Generates responses using text-to-speech

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

- Provides audible feedback to the user

This architecture ensures smooth interaction, scalability, and efficient system performance.

B. Compilation

Compilation in this system refers to converting the Python-based program into executable instructions.

- The code is written in Python using libraries for AI and automation
- The program is tested for syntax and logical errors
- Execution is performed through Python interpreters in environments like VS Code or Jupyter Notebook

The compiled program performs the following functions:

- Speech recognition
- NLP-based command processing
- Task execution
- Response generation

The compiled firmware is uploaded to the ESP32 using USB, enabling the hardware to execute real-time energy.

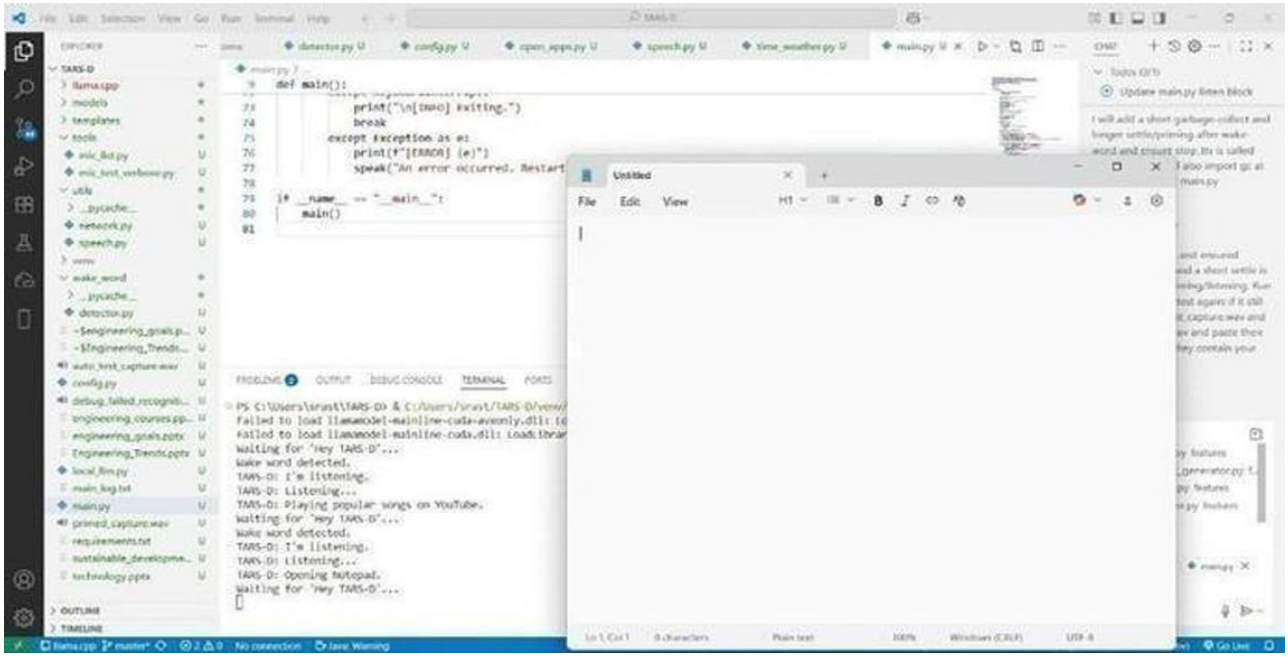

Fig1: Notepad opening using TARS-D AI Voice Assistant

C. *Training*

Although the system does not use traditional machine learning training, it involves system tuning and configuration for natural language models.

1. Language Model Fine-Tuning
   - Tuning the transformer-based model (e.g., BERT/GPT) for accurate intent analysis.
   - Refining conversational context preservation across multiple turns.
2. Speech Module Calibration
   - Adjusting microphone sensitivity for accurate voice capture.
   - Testing the voice-to-text conversion for variations in accent and tone.
3. System Integrity Testing
   - Verifying that individual modules of the TARS-D AI Voice Assistant function correctly.
   - Ensuring accurate processing of voice input, NLP interpretation, and command mapping.

D. *Deployment*

System deployment involves integrating the NLP processing, intent mapping, and backend logic into a functional desktop application. Deployment steps include:

1. Backend Processing Setup
   - Setting up the core decision-making and execution unit of the TARS-D AI Voice Assistant.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

2. Command Mapping Integration
   - Processing the user request using Python-based logic and control structures after intent is identified.
3. System-Level Library Linking
   - Interacting with various system and application-level libraries to carry out actions.
4. Functional Task Execution
   - Executing actions such as opening and closing applications, browsing the internet, managing files, and sending emails.
5. User Interface Finalization
   - Integrating the Text-to-Speech module for audible feedback to the user.

E. *Evaluation*

System evaluation assesses the performance and effectiveness of the developed voice assistant, based on the test objectives and features to be tested. The evaluation is based on the following criteria:

IV. Verification of Speech and NLP Processing

- Accurate conversion of voice input into text.
- Correct intent interpretation.

Command Mapping and Execution Performance

- Proper mapping of commands to system functions.
- Successful execution of system and web-related tasks.

V. Text-to-Speech Response Quality

1. Clear and correct voice output.
2. Validation of supported customization options.

System Robustness and Reliability

3. Performance consistency without frequent failures.
4. Graceful handling of invalid or unclear commands.

IV. SYSTEM IMPLEMENTATION

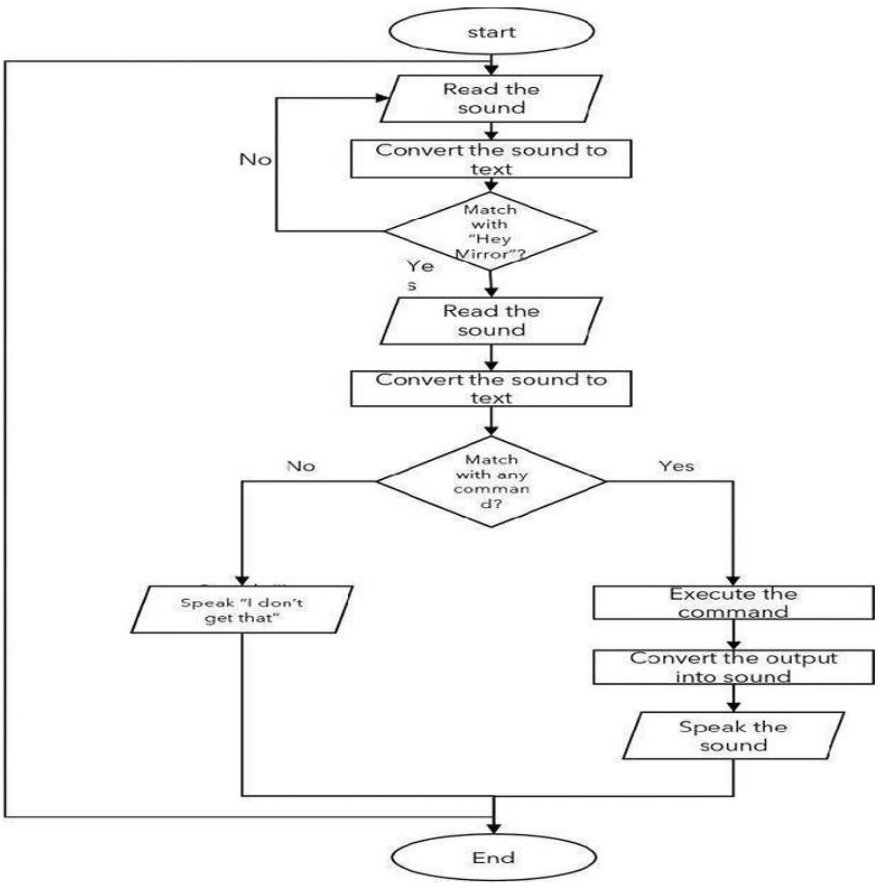

Fig 2: Activity diagram

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

The system implementation involves integrating the core modules of speech recognition, NLP processing, backend execution logic, and text-to-speech output into a functional, voice-driven desktop application.

A. Speech Input Acquisition

Speech input acquisition is the initial stage where the user provides commands through spoken language. The assistant captures audio input using a wired or Bluetooth microphone. This raw voice input forms the primary data source and must be clear, supporting different microphone types and involving noise handling to reduce background interference.

B. Speech Recognition

Speech recognition is the process of converting spoken language into digital text, which in this project is performed using APIs such as Google Speech Recognition. The module analyzes the audio signal and converts it into machine-readable text, handling variations in accent, tone, and pronunciation.

C. NLP Pre-processing

NLP preprocessing prepares the recognized text for intent analysis. Raw text is cleaned of filler words or inconsistencies using techniques like Tokenization (breaking the input sentence into individual tokens) and Stop Word Removal (eliminating unnecessary words like "please" or "the") to reduce processing overhead and improve analysis efficiency.

D. Intent Recognition and Command Mapping

Intent recognition identifies the purpose of the user's command by matching extracted keywords with predefined command patterns. This module determines whether the command is related to system control, media playback, communication, or web search, and then maps the command to the appropriate backend function.

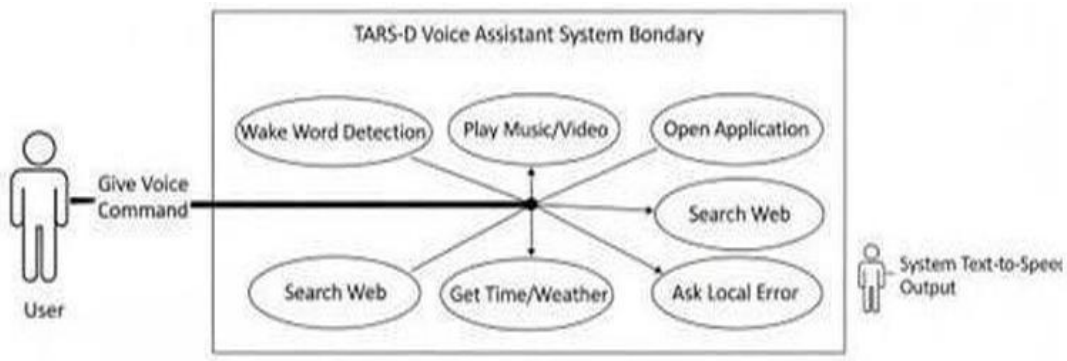

Fig 3: Use Case Diagram

E. Backend Processing

The backend processing module serves as the core decision-making and execution unit. Once the intent is identified, this module processes the request using Python-based logic and control structures. It interacts with various system and application-level libraries to carry out actions such as opening and closing applications, browsing the internet, managing files, and automating routine desktop operations.

F. Task Execution

Once the command is processed, the system executes the corresponding task. The execution may be offline (e.g., opening files or applications) or online (e.g., searching the web or sending emails). This phase handles all actions, including hands-free control, and must successfully confirm that the assistant has correctly interpreted the user's intent.

G. Response Generation

After task execution, the system generates a suitable textual response indicating the task status or requested information. This response is then converted into audible speech using a Text-to-Speech (TTS) engine, ensuring effective communication and completing the interaction cycle.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13441

V. CONCLUSION

The TARS-D AI Voice Assistant successfully demonstrates the implementation of an intelligent, desktop-based voice-controlled system using Natural Language Processing and speech recognition techniques. The system enables hands-free interaction, allowing users to perform various desktop and internet-based tasks efficiently through voice commands. By integrating speech recognition, NLP preprocessing, backend task execution, and text-to-speech modules, the assistant delivers accurate, responsive, and user-friendly interaction. The project emphasizes accessibility, particularly for visually impaired and mobility-impaired users, while maintaining data privacy through local processing. Overall, the system meets its functional objectives and proves to be a reliable and scalable solution for enhancing human-computer interaction in desktop environments.

VI. ACKNOWLEDGEMENT

We express our gratitude with great pleasure to Rajya Vokkaligara Sangha for the support and motivation extended to us in all aspects during the completion of the project.
We express our heartiest gratitude with great pleasure to Bangalore Institute of Technology, Bangalore that provided us an opportunity to fulfill our cherished desire to attain our goal.
We extend our thanks to our Principal Dr. Vijaya Prakash for encouraging us in all aspects to complete the project.
We immensely thank Prof. Vedasree T K, Assistant Professor, Dept. of ISE, for supporting and guiding us in carrying out our preparations for the project.
We would like to thank Dr. Asha T, Professor and HOD, Dept. of ISE, who continually helped us with her suggestions and ideas.
Finally, we thank one and all who have helped us directly or indirectly in the completion of the Project.

REFERENCES

[1]. Bisht, Vishal. "Desktop Voice Assistant System with the Help of Natural Language Processing." Grenze International Journal of Engineering and Technology, vol. 8, no. 2, June 2022, pp. 419-424. Grenze Scientific Society.
[2]. Pandey, Divisha, et al. "Voice Assistant Using Python and AI." International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 5, May 2022, pp. 832-838, www.irjet.net.
[3]. Pakhmode, Sonali, et al. "NLP Based AI Voice Assistant." International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 7, no. 3, Mar. 2023, pp. 1-9, doi:10.55041/IJSREM18521.
[4]. Pandey, Sri Meenakshi, et al. "AI Based Virtual Assistant." International Journal for Research Trends and Innovation (IJRTI), vol. 8, no. 3, 2023, pp. 100-106.
[5]. Bisht, Vishal. "Desktop Voice Assistant System with the Help of Natural Language Processing." Grenze International Journal of Engineering and Technology, vol. 8, no. 2, June 2022, pp. 419-424. Grenze Scientific Society.
[6]. Sharma, Rajat, and Adweteeya Dwivedi. "JARVIS" AI Voice Assistant." International Journal of Science and Research (IJSR), vol. 11, no. 5, May 2022, pp. 386-393, doi:10.21275/SR22503183839.
[7]. Indukuri, Manikanta Sai Varma, et al. "Voice Assistant Using Artificial Intelligence." International Journal of Engineering Development and Research (IJEDR), vol. 10, no. 2, 2022, pp. 105-112, www.ijedr.org.
[8]. Yadav, Ankush, et al. "Desktop Voice Assistant for Visually Impaired." International Journal of Recent Technology and Engineering (IJRTE), vol. 9, no. 2, July 2020, pp. 36-39, doi: 10.35940/ijrte.A2753.079220.
[9]. "The Pandey, Divisha, Afra Ali, Shweta Dubey, Muskan Srivastava, Shyam Dwivedi, and Md. Saif Raza. "Voice Assistant Using Python and AI." International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 5, May 2022, pp. 832-838, IRJET.net.
[10]. Basu, Indranil, et al. "Python-Based Desktop Assistant with Voice Recognition." International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 3, no. 11, May 2023, pp. 398-403, doi: 10.48175/IJARSCT-10618.

