Kalpa Publications in Computing
Volume 19, 2024, Pages 73–81
Proceedings of 6th International Conference on Smart Systems and Inventive Technology

Voice Recognition System with STORJ Decentralized Storage

S. L. Jany Shabu[1], Pooja R A[2], Ronie Joe J[3], J. Refonaa[4], Praveen.S[5], R.Abirami[6]
Sathyabama Institute of Science and Technology, Chennai, India
janyshabu.cse@sathyabama.ac.in, poojaamarnath1533@gmail.com, ronniejoe_j@icloud.com, refonna.cse@sathyabama.ac.in, praveenbca63@gmail.com, abirami.r.cse@sathyabama.ac.in

Abstract

The proposed Virtual Assistant System (VAS) represents a new solution revolutionizing user interactions with technology by seamlessly integrating Natural Language Processing (NLP) and Artificial Intelligence (AI), facilitating effortless communication through voice commands. By employing cutting-edge speech recognition algorithms, the proposed system accurately translates the voice input into text, adapting responses based on individual user preferences over time. The proposed system offers a diverse range of functionalities including information retrieval, task automation, and smart home control to assist users in managing the tasks hands-free with an intelligent interface providing varying levels of technical expertise. Safeguarding user privacy and control, the system allows users to opt-in or opt-out of data collection with complete transparency and robust security measures. Continuous improvement through extensive testing and user feedback addresses the challenges like accurately interpreting complex commands, positioning the Virtual Assistant System (VAS) as a sophisticated, personalized, and privacy-aware solution at the forefront of virtual assistant technology.

Keywords: Artificial Intelligence (AI), Data Analysis, Virtual Assistance, User Privacy, Automation, Smart Home

1 Introduction

The integration of Artificial Intelligence (AI) with machines showcases the ability to mimic human-like thinking. This involves designing a computer system that typically requires interaction from humans. Python, as a language, facilitates the development of a voice assistant. The instructions for the voice assistant can be customized according to user requirements. Python offers the Speech Recognition API, allowing the conversion of speech into text, a functionality seen in popular voice assistants like

Alexa and Siri. Creating a personalized assistant in Python presents an interesting task, enabling users to perform daily tasks such as sending emails, searching on Google, playing music, or opening specific applications through voice commands. Speech recognition emerges as an effective and natural way for people to interact with applications, offering an alternative to traditional input devices like mouse and keyboards. Its hands-free nature allows users to be productive and stay informed in situations where other interfaces may not be convenient. Speech recognition finds utility in various applications and environments in daily life. Beyond general interaction, it plays a significant role in assisting individuals with functional disabilities, providing a more accessible means of communication. In recent years, cloud storage systems have gained widespread adoption to meet the data storage and sharing needs of companies, organizations, and individuals. However, entrusting important and private data to centralized systems raises concerns about security. Centralized systems are susceptible to cyberattacks and service interruptions, prompting a progress towards decentralized data storage. Incidents like the Facebook Cambridge Analytica scandal have fueled a strong shift from centralized to decentralized data storage systems. This transition aims to enhance data security and mitigate the risks associated with centralized storage.

2 Literature Survey

Dr. Rand Hindi, a notable entrepreneur recognized by Forbes Under 30 [1], has voiced concerns regarding data privacy within AI-powered voice assistants. His company, Snips, founded in 2013, specializes in developing voice technology with a focus on privacy by design. J. Iso-Sipila et al. [2] presented a novel approach to multi-lingual speaker-independent Voice User Interfaces (VUI) for mobile devices. Their system supports speech recognition and synthesis in over 40 languages, even on resource-constrained embedded devices. Jinmook Lee et al. [3] introduced an energy-efficient Speech Extraction (SE) processor to enhance speech recognition accuracy in Head-Mounted Display (HMD) systems. Their implementation showcases significant improvements in speed and memory usage for low-power applications.

Laura BURbach et al.'s study [4] on acceptance factors of virtual voice assistants reveal user concerns about privacy and data handling. The research highlighted privacy features as a critical determinant of user acceptance. Pankaj Kunekar et al. discussed about the advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) leading to the widespread adoption of voice assistants [5]. Their study emphasized the evolution of voice assistants from smartphones to home automation systems. Peng Cheng et al. conducted a survey on the security and privacy aspects of Personal Voice Assistants (PVAs) [6]. With the increasing popularity of PVAs, the research underscored the need for robust security measures to address user and policy concerns.

Saqib Ali et al. proposed a domain-specific Intelligent Personal Assistant (IPA) capable of processing bilingual voice commands [7]. Their innovative approach leveraged Finite State Automation (FSA) to train the system with limited data, making it proficient in Natural Language Processing (NLP). The Sphinx-4 framework, while widely used for speech recognition, has limitations such as sensitivity to background noise and struggles with non-native accents [8]. Users should consider these constraints when selecting speech recognition frameworks. Saqib Ali et al. explored the potential of blockchain technology for decentralized record management [9]. They highlight blockchain's security, immutability, and fault tolerance, making it suitable for various applications, including financial transactions and identity management.

3 Existing System and its Challenges

A. Data Security

Data security emerges as a concern with Personal Voice Assistants (PVAs). These systems are constantly listening and recording, which raises privacy issues. Here are some key considerations:

- Data Storage and Retention: Voice assistance providers must be transparent about how long they store user data and allow users to delete their data.
- Data Encryption: All data transmitted between the voice assistant and the cloud servers should be encrypted to prevent interception by malicious actors.
- Anonymization: Personal information should be anonymized to reduce the risk of personal identification.
- User Consent: Individuals should be given the option to either participate in or abstain from data collection, and they should be informed about the specific data being collected and the purposes for which it is being collected.

B. Emotion Recognition

Emotion recognition is a capability some voice assistants are about to incorporate. Here are some aspects to consider:

- Ethical Use: Emotion recognition should be used ethically, and user consent is vital. It has the potential for misuse, such as invasive monitoring.
- Accuracy and Bias: Ensuring that emotion recognition is accurate across different demographics and doesn't introduce or perpetuate biases. This is the most challenging task.
- Privacy: The voice assistant should recognize emotions based on voice tone and content without violating user privacy by trying to analyze emotions directly.

C. Ethical and Bias Concerns

Ethical and bias concerns are crucial in the development and deployment of voice assistants:

- Bias in Training Data: Training data used for voice assistants can contain biases. Developers need to actively work on reducing and mitigating these biases.
- Diverse Representation: Efforts should be made to include diverse voices, backgrounds, and experiences in the development process to ensure inclusivity.
- Transparency: Users should know how voice assistants work, what data is collected, and how decisions are made to maintain transparency.

- User Control: Providing users with control over what their voice assistants say or do, allowing them to customize responses within ethical boundaries.

4 Proposed System

The proposed voice assistant system integrates cutting-edge technologies to offer seamless functionality, robust security, and efficient access control. Leveraging Storj for data security ensures that sensitive user information, including personal preferences, voice recordings, and interaction history, is securely stored on a decentralized network, safeguarding it against unauthorized access and data breaches. This voice assistant system is equipped with a comprehensive set of features, including the ability to read PDF files, compose and send emails, execute user commands, and manage system shutdown procedures. Through advanced Natural Language Processing (NLP) algorithms, the system can accurately interpret user requests and execute corresponding actions with precision and speed. Access control mechanisms are implemented to ensure that only authorized users can interact with the data of the voice assistant system. User authentication protocols, such as biometric authentication or password-based authentication, can be employed to verify user identities before granting access to sensitive functionalities and data. By integrating Storj for data security and access control measures, this voice assistant system enhances the privacy and confidentiality of the user's data while delivering a seamless and intuitive user experience.

5 Methodology

The development of a voice assistant system will adhere to a structured methodology leveraging Python as the primary programming language. The project will follow an iterative approach, combining elements of Agile and Waterfall methodologies to ensure flexibility and adaptability while maintaining a systematic progression.

5.1 Requirement Analysis

The project will begin with a comprehensive analysis of user requirements and functional specifications. This phase involves gathering insights into user expectations, desired features, and system functionalities that encompass reading PDF files, sending emails, executing commands, and managing system operations like shutdown.

5.2 Design Phase

Following requirement analysis, the proposed system architecture and design has been conceptualized. Python's versatility has been utilized to design modular components, ensuring scalability and maintainability. Design decisions have also considered integration points with Storj for enabling data security and access control mechanisms.

5.3 Implementation

The core functionality of the voice assistant system will be developed using Python. Libraries and frameworks such as NLTK (Natural Language Toolkit) for natural language processing and SMTP libraries for email handling will be integrated. Storj APIs will be utilized for secure data storage and retrieval. Continuous integration and version control practices will be employed to streamline the development processes.

The proposed methodology mainly focuses on development of a voice assistant with the integration of decentralization to achieve data privacy and provide control of the user's data to them. For this, all the data collected will be stored and retrieved from storj blockchain. Some other sort of blockchain can also be implemented in the future versions.

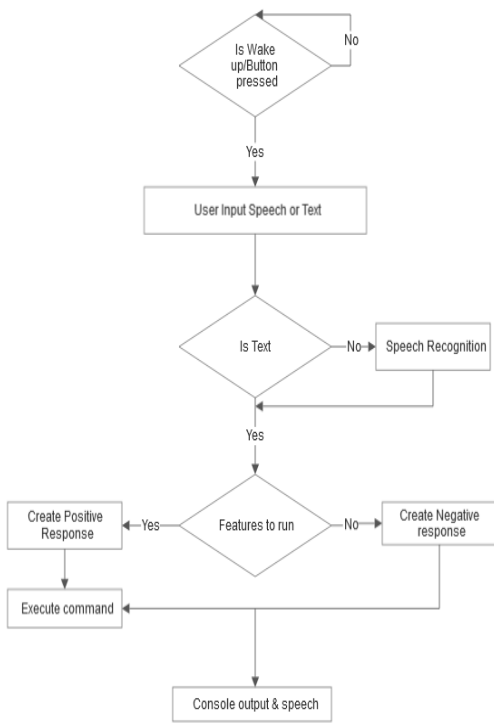

Figure 1 Flow Chart for Voice Recognition System

The flowchart (Figure 1) begins with the "Start" symbol, indicating the initiation of the voice assistant system. The user starts the interaction with the voice assistant by activating it, typically through a wake word or a button press. The system captures the user's spoken command as speech input using a microphone. The speech input is processed through a speech recognition module, converting spoken words into text. The system identifies the user's intent by analyzing the converted text. This involves understanding the user's request or command. Based on the recognized intent, the system processes the command to determine the appropriate action. This could involve retrieving information, performing tasks, or controlling connected devices. The system executes the desired action, which may include querying databases, accessing APIs, or triggering specific device functions. After executing the action, the voice assistant generates a response, typically in the form of spoken words. The generated response is converted into speech using a text-to-speech synthesis module. The synthesized speech is given as output to the user through speakers or headphones, providing a voice response to the user's initial command. The system may capture user feedback or additional commands for further interaction. The flowchart concludes with the "End" symbol, indicating the completion of the interaction. This flowchart provides a high-level overview of the key steps involved in the voice assistant system, from user initiation to system response. Depending on the complexity of your system, you may need to add more detail or additional branches to capture various scenarios and functionalities. The working of Decentralized File Storage is shown in Figure 2.

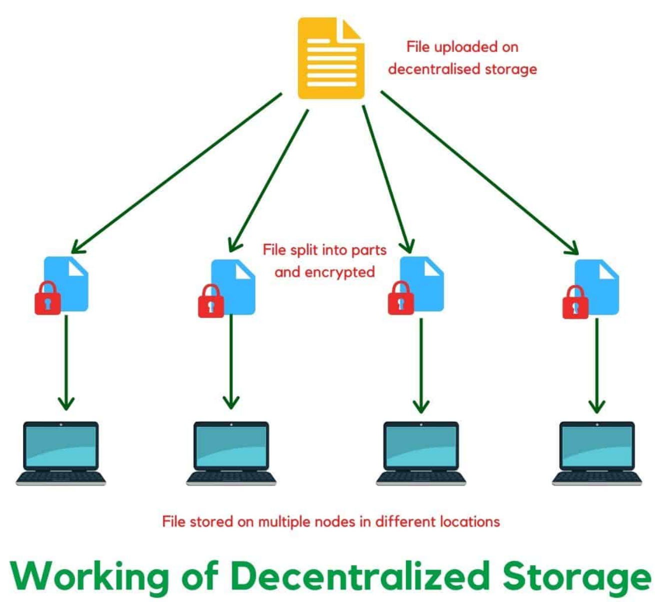

Figure 2: Working of Decentralized File Storage

A. Implemented Modules

Speech Recognition Module is crucial for a voice assistant, this module enables the system to recognize and understand the user's voice. The installation command for this module is `pip install SpeechRecognition`. Next, the DateTime Module utilized the built-in date and time package to display the current date and time. Third is the Wikipedia Module, which leverages Wikipedia as a vast knowledge source for information retrieval and search. Its Installation command is `pip install wikipedia`. Fourth module is the Webbrowser Module for performing web searches directly from the application. The fifth module is OS Module, which is a part of Python's standard utility modules, it facilitates interaction with the operating system. The sixth module is the Pyaudio Module, which includes Python bindings for PortAudio, enabling audio input and output functionality.

Next, is the PyQt5 Module, which includes comprehensive Python bindings for Qt v5, supporting application development. Its installation command is `pip install PyQt5`. Followed by that Python Backend module is integrated to receive output from the speech recognition module to identify API calls and context extraction, and return the required output. After that, the Text to Speech Module (Pyttsx3) is integrated to convert written text to phonemic representation and generate sound waveforms for text-to-speech output. This is followed by the Speech to Text Conversion process, which uses speech recognition to convert the spoken input into textual format, making it understandable to the computer. Finally, the textual output decodes voice commands, performs operations, and displays the voice command as textual output in the terminal.

These modules collectively enable the voice assistant to understand user commands, retrieve information, interact with the web, and provide textual and speech-based responses.

B. Decentralization Modules

Storj is a decentralized cloud storage platform used here to allow users to store their data in a secure and decentralized manner. It uses a network of nodes to store encrypted fragments of data and provides a marketplace where users can buy and sell storage space.

The above module enables the voice assistant to store data in a decentralized storage which is protected by smart contract access control to enable only the authorized user group can access the voice data.

6 Results and Discussion

The implementation of the voice assistant system using Python has resulted in a highly functional and efficient platform capable of seamlessly executing a range of tasks, including reading PDF files, composing and sending emails, responding to user commands, and managing system shutdown procedures. Leveraging Python's extensive libraries and frameworks, the system demonstrated robust Natural Language Processing (NLP) capabilities, accurately interpreting user inputs and executing corresponding actions with precision. The iterative development methodology facilitated the optimization of the system, ensuring reliability, accuracy, and performance. Through testing, the system has been thoroughly validated, demonstrating its effectiveness in real-world scenarios. Validated outcomes are shown in Figure 3 -Figure 4.

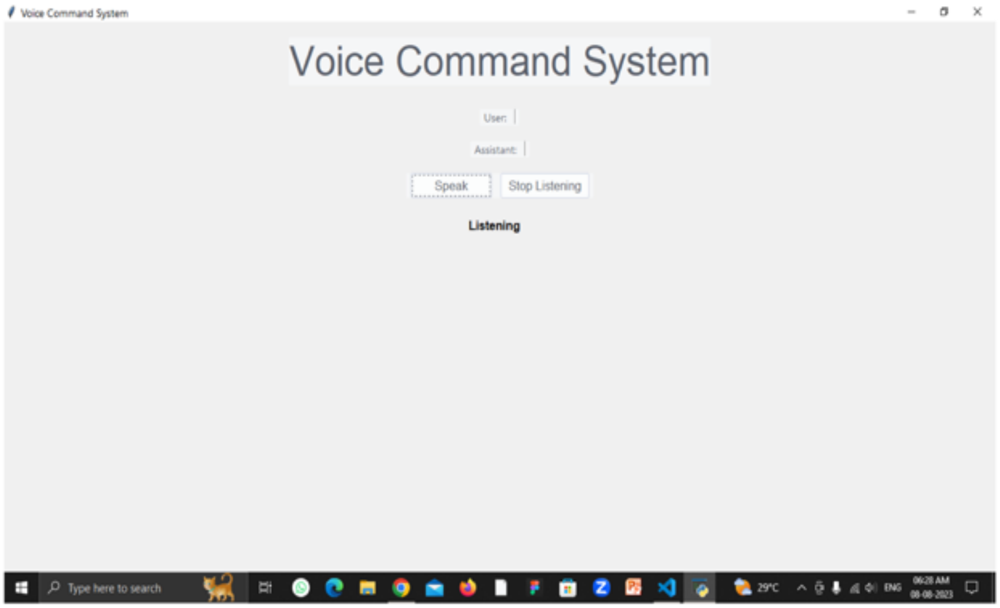

Figure 3: Voice Command System Output

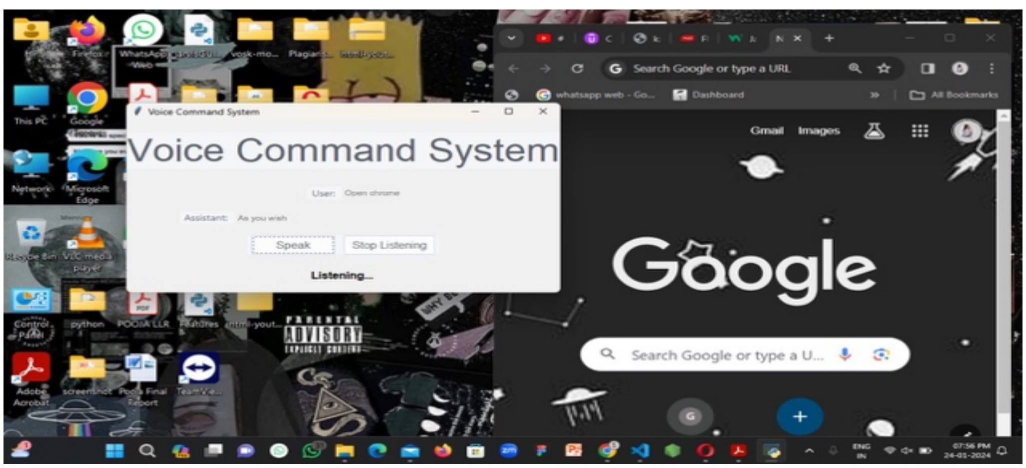

Figure 4: Information Gathering Output

7 Conclusion

In conclusion, the development of the voice assistant system utilizing Python as the major programming language, coupled with integration with Storj for data security, represented a significant step towards delivering a robust, user-friendly, and secure solution. By leveraging Python's versatility and the extensive ecosystem of libraries, this study has successfully implemented a system capable of efficiently handling various tasks, including speech recognition, PDF manipulation, email management, and system control. The integration of Storj ensures that user data is securely stored and accessed, addressing concerns related to privacy and data security. Through an iterative development process, we have meticulously designed, implemented, and tested the system to meet user requirements and quality standards. Moving forward, ongoing maintenance, updates, and enhancements will be crucial to ensure the system remains responsive to user needs and aligned with evolving technological advancements.

References

[1] https://www.forbes.com/sites/rachelwolfson/2018/09/14/blockchain-based-ai-voice-assistant-brings-data-privacy-to-smart-homes/?sh=340851ad6b50

[2] J. Iso-Sipila, M. Moberg, and O. Viikki, "Multi-lingual speakerindependent voice user interface for mobile devices," in ICASSP 2006. IEEE, 2006, vol. I, pp. 1081-1084.

[3] Jinmook Lee, Seongwook Park, Injoon Hong and HoiJun Yoo, “An Energy-efficient Speech Extraction Processor for Robust User Speech Recognition in Mobile Head-mounted Display Systems”, IEEE Transactions on Circuits and Systems II, April 2017.

[4] Laura BURbach, Patrick Halbach, Nils Plettenberg, Johannes Nakyama, Matrina Ziefle, Andre Calero Valdez, “Ok google, Hey Siri, Alexa, Acceptance relevant of virtual voice assistants” I, IEEE 2019. (ICCIT) Dec. 2010.

[5] Merkle R C. Protocols for public keycryptosystems[C]//Security and Privacy, 1980 IEEE Symposium on. IEEE, 1980: 122-122.

[6] Pankaj Kunekar; Ajinkya Deshmukh; Sachin Gajalwad; Aniket Bichare; Kiran Gunjal; Shubham Hingade, “AI-based Desktop Voice Assistant”, 2023 IEEE 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), 20-21 Jan. 2023

[7] Peng Cheng, Utz Roedig, “Personal Voice Assistant Security and Privacy—A Survey”, Proceedings of the IEEE ,Volume: 110 Issue: 4, 11 March 2022.

[8] Philipp Aichinger, Martin Hagmuller, Berit SchneiderStickler, Jean Schoentgen and Franz Pernkopf, “Tracking Of Multiple Fundamental Frequencies in Diplophonic Voices”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Volume: 26, Issue: 2, October 2017.

[9] R. C. Merkle, “Protocols for public key cryptosystems,” in 1980 IEEE Symposium on Security and Privacy, April 1980, pp. 122–122.

[10] Saadman Shahid Chowdury, Atiar Talukdar, Ashik Mahmud, Tanzilur Rahman, “Domain specific Intelligent personal assistant with bilingual voice command processing,”IEEE 2018

[11] Saqib Ali, Guojun Wang, Bebo White, Roger Leslie Cottrell, “A Blockchain-Based Decentralized Data Storage and Access Framework for PingER”, IEEE 06 September 2018.

[12] W. Matthews and L. Cottrell, “The pinger project: active internet performance monitoring for the henp community,” IEEE Communications Magazine, May 2000.