Educational Administration: Theory and Practice
2024, 30(5), 10967-10974
ISSN: 2148-2403
https://kuey.net/

Research Article

Personal AI Voice Assistant

Prof. (Dr.) Manisha Rajesh Mhetre[1*] , Bhushan Deshmukh[2], Parshvnath Bandrewar[3], Rohit Bodhak[4], Amar Chavan[5]

[1*]Department of Instrumentation and Control EngineeringVishwakarma Institute of Technology Pune, India. manisha.mhetre@vit.edu
[2]Department of Instrumentation and Control Engineering Vishwakarma Institute of Technology Pune, India bhushan.deshmukh21@vit.edu
[3]Department of Instrumentation and Control Engineering Vishwakarma Institute of Technology Pune, India parshvnath.bandrewar21@vit.edu
[4]Department of Instrumentation and Control Engineering Vishwakarma Institute of Technology Pune, India rohit.bodhak21@vit.edu
[5]Department of Instrumentation and Control Engineering Vishwakarma Institute of Technology Pune, India amar.chavan21@vit.edu

Citation: Prof. (Dr.) Manisha Rajesh Mhetre, et al (2024), Personal AI Voice Assistant , *Educational Administration: Theory and Practice*, *30*(5), 10967-10974
Doi: 10.53555/kuey.v30i5.4869

ARTICLE INFO

ABSTRACT

Artificial intelligence technologies are increasingly being used in human life, with the Internet of Things (IoT) facilitating the integration of smart devices into social networks. One trend in AI is recognizing human language, leading to new means of natural human-machine interaction. One such tool is voice assistants, which can be integrated into various intelligent systems. This paper discusses the principles of voice assistants, their limitations, and the method of creating a local voice assistant without cloud services. Intelligent Voice Assistants (IVA) like Siri and Alexa are created to assist users with simple digital tasks. voice-operated IVA that can process direct commands in English ,performing menial tasks for users. The language processing is performed by a modified finite state automaton, utilizing the subject/action structure of commands to reduce the word domain size.

Keywords— Speech Recognition, Face Recognition, TTS, Voice command, Voice assistant.

I. Introduction

The integration of artificial intelligence (ai) technologies with the internet of things (iot) is revolutionizing human-machine interactions. voice assistants, exemplified by mainstream solutions like siri and alexa, offer intuitive interfaces for users. however, their heavy reliance on cloud services raises concerns regarding privacy, latency, and accessibility. in response, our research focuses on the development of a local voice assistant capable of processing commands in english and hindi. we utilize a modified finite state automaton for efficient language processing, exploiting command structures to reduce the word domain and enhance accuracy. furthermore, a generalization function enables the assistant to understand multiple languages without extensive modification, ensuring adaptability in diverse linguistic environments. by offering a local solution, we aim to address privacy concerns and improve accessibility in iot ecosystems, particularly in scenarios with limited cloud connectivity.

I have concentrated my study on creating a local voice assistant that can understand and execute commands in Hindi and English as a solution to these difficulties. By preventing sensitive speech data from being sent to the cloud and keeping it on the user's device, this local processing method seeks to allay privacy concerns. Furthermore, by doing calculations locally, the system can lower latency, resulting in quicker reaction times and an improved user experience.

I use a modified finite state automaton (FSA) for efficient language processing, which enables fast command execution and recognition. By using the organized character of orders, the FSA aims to decrease the word domain and enhance the voice assistant's precision and velocity.

II. RELATED WORK

1.Amazon Alexa and Echo Devices:

The integration of Amazon Alexa with Echo devices has revolutionized the way users interact with technology in their homes. Alexa's sophisticated architecture leverages advanced natural language understanding (NLU) algorithms and machine learning models to accurately interpret user commands and execute tasks seamlessly. Research into Alexa's development highlights the evolution of its capabilities, from basic voice recognition to sophisticated contextual understanding and personalized responses. By examining technical documentation and research papers on Alexa's architecture and functionality, insights can be gained into the underlying mechanisms driving its success in the market.

2.Microsoft Cortana Integration:

Microsoft Cortana stands as a prominent example of voice assistant integration within the Windows ecosystem, offering users intuitive voice-driven interactions across various Microsoft products. Cortana's journey from its inception to its current state underscores significant advancements in speech recognition and natural language processing (NLP). Through the analysis of Cortana's architecture and implementation details, researchers can gain valuable insights into the challenges and innovations driving the development of this voice assistant platform. Technical articles and research papers elucidating Cortana's capabilities and strategies for improving user engagement provide valuable context for understanding its impact on the digital landscape.

3.Google Assistant Development:

The development of Google Assistant has reshaped the landscape of voice-enabled technology, with its integration across Android devices, smart speakers, and other Google products. Leveraging cutting-edge machine learning techniques, Google Assistant boasts robust speech recognition capabilities and natural language understanding, enabling intuitive user interactions. Researchers exploring Google's advancements in voice technology can delve into research publications and developer documentation to uncover the methodologies and algorithms powering Google Assistant's functionality. Insights gleaned from these resources shed light on the evolution of voice assistant technology and its implications for future applications.

4.Blind Source Extraction for Speech Enhancement:

Blind source extraction (BSE) techniques play a crucial role in enhancing the quality of speech signals by isolating desired audio sources from noisy environments. Through preprocessing steps and advanced algorithms, BSE algorithms contribute to improving the signal-to-noise ratio (SNR) in multichannel speech data, thereby enhancing the performance of speech recognition systems. Research papers and studies focusing on BSE methodologies offer valuable insights into the challenges and innovations in speech enhancement, providing researchers with foundational knowledge for developing robust voice assistant systems capable of operating in diverse acoustic environments.

5.Automatic Speech Recognition Architectures:

Architectures for automatic speech recognition (ASR) and voice activity detection (VAD) continue to evolve, driven by the demand for enhanced accuracy, programmability, and scalability. Researchers investigating ASR architectures explore various approaches, including the utilization of deep neural networks (DNNs) and parallel processing units to optimize performance. Studies outlining architectures for ASR and VAD systems provide valuable guidance for designing efficient and scalable voice assistant solutions. By examining research by Swamy et al. and other experts in the field, researchers can gain insights into the latest advancements and methodologies shaping the future of ASR technology.

III. PROPOSED SYSTEM

The suggested system is a smart assistant called Jarvis that may perform a number of functions including sending emails, running local apps, displaying the current time, accessing webpages, and conducting Wikipedia searches. The system uses a number of Python libraries to do text-to-speech conversion, voice recognition, and web surfing.

The Text-to-Speech Engine (pyttsx3) is the part that converts text replies into speech, enabling voice communication between the user and the assistant. The pyttsx3 library serves for initialization and assembled and its speech settings are adjusted to produce a realistic-sounding interface. Speech Recognition (speech_recognition): This component records and interprets user-provided audio input that comes from the microphone. In order for the system to comprehend and respond to user instructions, spoken language must be converted into text using the SpeechRecognition library.

Wikipedia Integration (wikipedia): With the help of this component, the assistant can access summaries from the web page according to user requests. The system may seek and obtain relevant data by using the Wikipedia library. The user is then presented with the information read aloud.

Web Browsing (webbrowser): The webbrowser module allows the assistant to open a variety of websites, including Google, YouTube, and others and Stack Overflow. The user's ability to swiftly access to frequently used online pages is improved by this capability.

IV. SYSTEM ARCHITECTURE

User Interface (UI):
The UI component serves as the interface between the user and the voice assistant. It can be implemented as a mobile application or a standalone device with a microphone.

Speech Recognition Module:
This module is responsible for converting spoken language input from the user into text. It utilizes automatic speech recognition (ASR) techniques to accurately transcribe the user's commands into a machine-readable format.

Natural Language Understanding (NLU):
The NLU component interprets the text input received from the speech recognition module and extracts the user's intent and relevant entities from the command.This involves tasks such as entity recognition, intent classification, and semantic parsing.

Language Processing Module:
This module processes the interpreted command using a modified finite state automaton. It is designed to handle commands in two languages: English and Hindi. By utilizing a subject/action structure and a generalization function, it reduces the complexity of language processing and enables support for multiple languages without extensive modification.

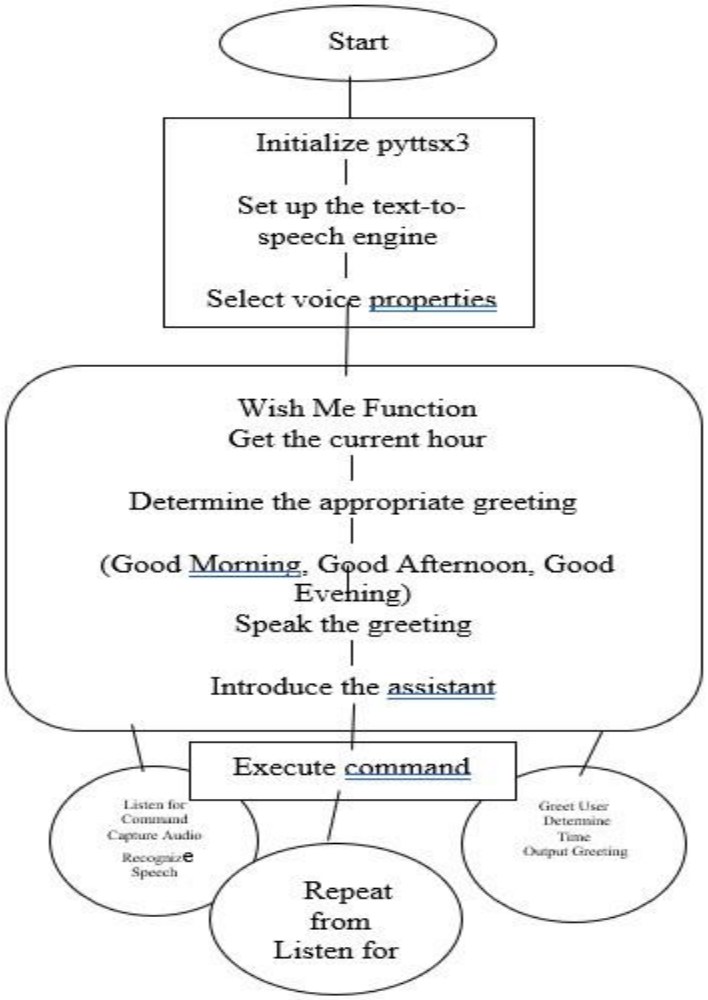
Action Execution Engine:
Once the user's intent and entities are identified, the action execution engine determines the appropriate action to take based on the command. It interfaces with various services and devices to perform tasks requested by the user, such as setting reminders, playing music, or controlling smart home devices.

Local Processing and Storage:
To ensure privacy and reduce reliance on cloud services, the voice assistant's processing and storage capabilities are localized. This allows the system to function without constant internet connectivity and minimizes the transmission of sensitive user data over the network.

Feedback and Response Generation:
After executing the requested action, the voice assistant provides feedback to the user to confirm successful completion or inform about any errors encountered during the process. This feedback can be generated through speech synthesis or displayed as text on the user interface.

Continuous Learning and Improvement:
The voice assistant system incorporates mechanisms for continuous learning and improvement. It can collect user feedback and usage data to refine its language processing capabilities, enhance accuracy, and adapt to user preferences over time.

Fig. 1: system flow chart



V. ALGORITHM

The proposed algorithm for the personal AI voice assistant project embodies a multifaceted approach to efficiently process user commands in English and Hindi while prioritizing privacy, accessibility, and performance. Beginning with the capture of speech input, the algorithm swiftly transitions into the realm of automatic speech recognition, where spoken language is deftly transcribed into text format. This textual representation then undergoes rigorous scrutiny through the lens of natural language understanding (NLU), where the user's intent and pertinent entities are meticulously extracted. Key to the algorithm's versatility is its adeptness at language detection, seamlessly discerning between English and Hindi inputs to facilitate language-specific processing. Central to its processing prowess lies a modified finite state automaton (FSA), ingeniously engineered to exploit the subject/action structure inherent in commands, thereby streamlining language processing complexity. Leveraging a generalization function, the algorithm ensures harmonious coexistence with both English and Hindi commands, underscoring its adaptability in multilingual contexts. Action determination, executed with precision, culminates in local action execution, minimizing reliance on cloud services and upholding user privacy by confining sensitive data within the device. Through meticulous feedback generation, the voice assistant earnestly communicates the outcome of executed actions, fostering user confidence and engagement. Continual learning mechanisms underpin the algorithm's evolution, gathering user feedback and usage data to refine language processing prowess iteratively. Robust error handling, offline mode support, and efficient resource utilization are woven into its fabric, fortifying its reliability, versatility, and resilience. With modularity at its core, the algorithm remains primed for future enhancements and extensions, ensuring its relevance in an ever-evolving landscape. Moreover, a commitment to compatibility, performance optimization, and accessibility underscores its commitment to inclusivity and user-centricity. In its holistic design, the algorithm represents a sophisticated synthesis of cutting-edge technology and user-centric design principles, poised to empower users with intuitive, efficient, and personalized voice assistant interactions.

VI . METHODOLOGY

The methodology employed in this personal AI voice assistant project encompasses several key steps aimed at creating a functional and efficient system for natural language interaction and task automation.
Firstly, the project initializes with loading necessary libraries and setting up the speech recognition and text-to-speech engines. This includes importing essential modules such as speech_recognition, pyttsx3, and datetime for handling voice input, output, and time-related functionalities, respectively.
Next, the project proceeds with defining functions for voice output (speak()) and initializing greetings (wishMe()). These functions are crucial for providing a personalized and interactive experience to the user, ensuring smooth communication between the user and the AI assistant.

The core functionality of the assistant lies in its ability to interpret user commands. This is achieved through the takeCommand() function, which utilizes the speech recognition module to convert spoken commands into text format. The assistant then processes these commands to perform various tasks based on predefined triggers and keywords.

The assistant offers a range of functionalities, including searching **Wikipedia for information, opening web browsers (e.g., YouTube, Google Chrome, Gmail), retrieving weather forecasts, displaying current time, providing news headlines, capturing photos, and performing web searches.**

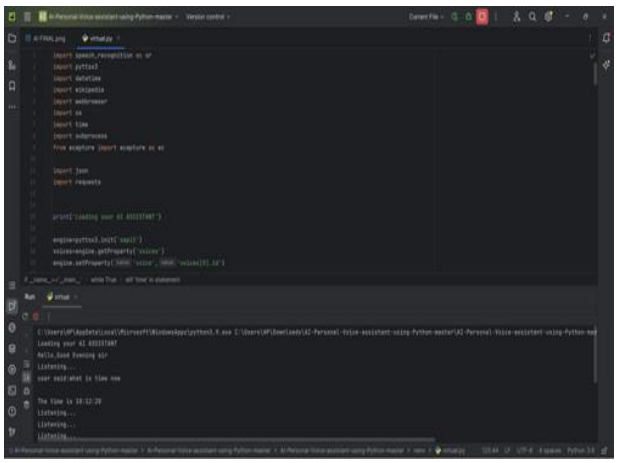
Furthermore, the project incorporates external **APIs** and web scraping techniques to fetch real-time data, such as weather information and news headlines, enhancing the assistant's capabilities and usefulness.

To ensure robustness and user satisfaction, error handling mechanisms are implemented to handle exceptions gracefully and provide appropriate feedback in case of unrecognized commands or errors during execution.

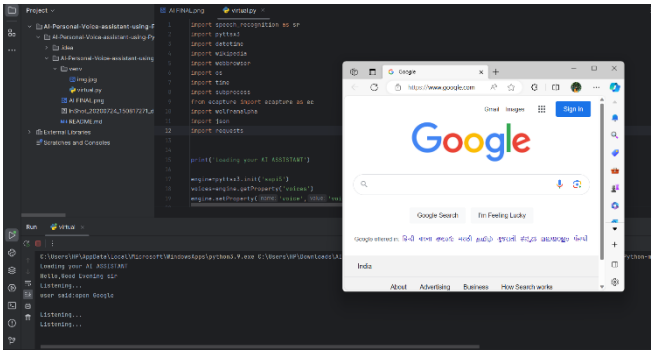
The project follows an iterative development approach, allowing for continuous testing, evaluation, and improvement based on user feedback and emerging requirements. Regular updates and enhancements are made to the assistant's functionalities and performance to adapt to evolving user needs and technological advancements.

Overall, the methodology employed in this project aims to create a versatile and user-friendly AI voice assistant capable of effectively assisting users with various tasks through natural language interaction, thereby enhancing productivity and convenience in daily life.

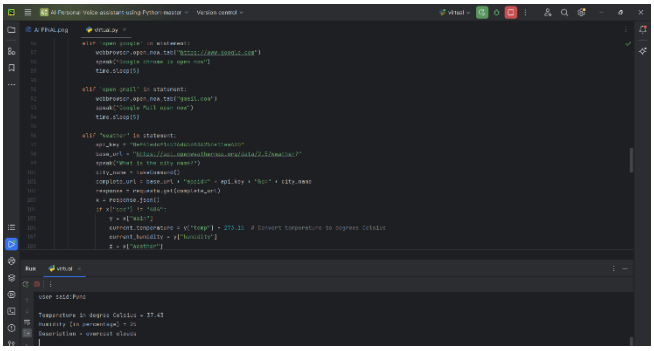
Fig. 2 : ASR process



The commands are listening by the project AI



The google is open after Asking for that



Telling the weather of pune city

VI. Implementation of ASR

Comprehending Automatic Speech Recognition (ASR):

This technique, known as Automatic Speech Recognition (ASR), turns spoken words into text. A number of intricate procedures and technological tools are needed to accomplish precise and effective voice recognition. Within the given code, speech_recognition library handles ASR mostly by recognising spoken words using Google's Web Speech API. Let's take a closer look at the ASR theory and its application in these kinds of systems.

Important Elements of ASR Acoustic Modelling:

The link between audio impulses and phonetic units in speech is represented by acoustic models. To learn the numerous ways that phonemes, which are discrete units of sound, are uttered in various settings, these models are trained on extensive datasets of audio recordings and their related transcriptions.

Modelling Languages:

Language models forecast the possibility of a word order. By offering context-aware predictions, they aid in context comprehension and can greatly increase speech recognition accuracy. For instance, a language model makes sure that "door" is identified as a more likely word after "open the" in the phrase "open the door."

Dictionary Phonetic:

Words and their phonetic transcriptions are mapped in a phonetic dictionary. This aids in the deconstruction of words by the ASR system into their component phonemes, which may subsequently be compared to the acoustic model.

Feature Deletion:

In order to extract valuable features that can be applied to recognition, feature extraction entails processing the raw audio data. Mel-Frequency Cepstral Coefficients (MFCCs), which depict the audio signal's short-term power spectrum, are a frequently used characteristic.

Interpretation:

To translate the retrieved features into text, the decoding procedure integrates the phonetic dictionary, language model, and acoustic model. It entails looking over every transcription that may be made and choosing the one with the best chance.

How the Presented Code Initialization Uses ASR:

The recognizer and microphone instances are set up when the speech_recognition library is initialised. While the microphone records the spoken words, the recognizer analyses the audio input.

Searching for Audio:

The user can provide audio input to the machine. This entails using the microphone to record sound waves and transforming them into an audio format that the recognizer can understand.

Converting Speech to Text:

The audio recording is routed to Google's Web Speech API for automated speech recognition.

This includes:
Taking characteristics out of the audio stream is known as acoustic feature extraction.
Pattern recognition is the process of matching audio data to lexical and phonetic patterns using pre-trained models.
Language Understanding: Using language models to make sure the text that has been identified makes sense in its context.
Error Resolution:
When speech recognition fails (for example, because of background noise or ambiguous speech), the system has procedures in place to deal with the situation. If it cannot recognise the command, it asks the user to repeat it.
Using Google's Web Speech API Accuracy Has Many Benefits with the use of sophisticated machine learning models trained on enormous volumes of data, Google's Web Speech API can recognise a wide range of accents and languages with high accuracy.
Scalability:
Because the API is a cloud-based service, it can manage a lot of requests at once, which makes it appropriate for high-demand applications.
Usability:
A large portion of the complexity associated with ASR is abstracted by the API, which gives programmers an easy-to-use interface for incorporating speech recognition into their apps.
ASR Voice Assistants' Real-World Uses:
ASR is used by voice-activated assistants (VAAs) such as Siri, Google Assistant, and Alexa to comprehend and react to user orders.
Availability:
With voice-controlled interfaces, ASR technology helps people with disabilities by enabling hands-free device usage.
Translation Services:
Automated transcription services save time and effort by turning spoken content—such as lectures and meetings—into text instead of requiring human transcription.
Client Support:
ASR is used by interactive voice response (IVR) systems to comprehend consumer inquiries and deliver pertinent answers, improving the effectiveness of customer care.
Problems with Accents and Dialects in ASR:
Accurate recognition may be impacted by differences in dialects and accents. To adequately manage these variances, ASR systems must be trained on a variety of datasets.
Ambient Sound:
The audio input may be affected by noisy surroundings, which might make it difficult for the system to detect speech clearly.
Similar Phonemes:
Words with similar sounds but distinct meanings (such "two" and "too") can be difficult to interpret in context and require strong language models to distinguish.
Processing in Real Time:
For applications such as voice-controlled interfaces and live transcription, low-latency speech recognition is essential, requiring strong processing capabilities and effective algorithms.

VII. Literature survey

1. Voice based AI assistant

[1] In the paper titled "Voice based AI assistant," presented at The 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN) in the Philippines, advancements in voice recognition and natural language processing (NLP) are explored. The authors emphasize the growing role of virtual assistants in consumer technology, driven by improvements in speech recognition. The study focuses on the development of personal assistant software designed to simplify life by leveraging web-based semantic data sources, user-generated content, and comprehensive knowledge libraries. It highlights how AI programming can learn from user-provided data to better predict user needs. The paper notes the significance of speech recognition as a critical advancement in the functionality of smartphones and wearable devices. The research was added to IEEE Xplore on June 7, 2023, under the DOI: 10.1109/CICTN57981.2023.10141447, and was published by IEEE.

2. BRAIN – THE A.I.

[2] In the paper titled "BRAIN – THE A.I.," presented in The International Research Journal of Engineering and Technology (IRJET), the authors introduce a personal voice assistant that leverages user commands for task automation, aiming to enhance interaction efficiency and naturalness. This system incorporates unique face recognition technology to ensure that only authorized users can issue commands. BRAIN – THE A.I. is designed to handle a wide array of tasks, including delivering news updates, conducting searches, sending emails, playing games, setting reminders, providing location information, forecasting weather, and reading horoscopes. This

multifaceted functionality makes it a versatile and powerful tool for everyday use. The paper was added to IEEE Xplore on April 21, 2020**.**

3. Implementing voice commands for speech recognition.

[1] In the paper titled "Intelligent Personal Assistants (IPAs)," presented at The 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), the authors explore the development of software agents that perform tasks on behalf of humans, akin to chat bots. These IPAs interpret human speech and provide responses through synthesized voices. The paper proposes the implementation of speech recognition systems to create a virtual personal assistant, utilizing a local database, a SURR parser, and a synthesizer to enhance functionality and user interaction. This approach aims to improve the efficiency and responsiveness of virtual assistants. The research was added to IEEE Xplore on November 30, 2020, under the DOI: 10.1109/ICSCAN49426.2020.9262279, and was published by IEEE**.**

VIII. Future outcome

The future of this voice assistant project includes enhanced accessibility, increased productivity, personalized assistance, seamless integration with IoT ecosystems, improved natural language understanding, privacy and security enhancements, and expansion into diverse application domains. The voice assistant can accommodate users with disabilities and vision impairments, streamline workflows, and provide personalized recommendations. It can also serve as a central hub for managing interconnected smart devices, fostering trust and satisfaction. The voice assistant can implement robust encryption, authentication, and access control mechanisms to safeguard user data and ensure transparency. Its versatility allows it to be integrated into various application domains, such as education, healthcare, customer service, and entertainment. This project aims to transform the way users interact with technology and enrich their lives .

IX. CONCLUSION

In conclusion, this personal AI voice assistant project demonstrates the potential of artificial intelligence to revolutionize human-computer interaction and enhance daily life. By combining advanced natural language processing, task automation, and integration with various services and devices, the assistant offers users a convenient and efficient way to accomplish tasks and access information. The project's success highlights the importance of continual innovation and improvement in AI technologies to meet evolving user needs and preferences. Looking ahead, the future scope of this project is vast, with opportunities for further development in areas such as multilingual support, contextual understanding, and personalized user experiences. As AI continues to advance, the possibilities for enhancing the capabilities and usability of voice assistants like this one are virtually limitless. However, it's crucial to prioritize considerations such as data privacy, security, and inclusivity to ensure that these technologies benefit society as a whole. Ultimately, this project serves as a testament to the transformative potential of AI in reshaping how we interact with technology and navigate our increasingly connected world.

X. REFERENCES

1. Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”, published by No Starch Press, Second Edition, 2019.
2. Andreas Muller and Sarah Guido, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, published by O’Reilly, First Edition, 2016.
3. S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh, "Artificial Intelligence-based Voice Assistant", 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020.
4. K N., R. V., S. S. S. and D. R., "Intelligent Personal Assistant - Implementing Voice Commands enabling Speech Recognition", 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.
5. Shahid Chowdury, Atiar Talukdar, Ashik Mahmud, Tanzilur Rahman³Domain specific Intelligent personal assistant with bilingualvoice command processing´IEEE 2018.
6. Polyakov EV, Mazhanov MS, AY Voskov, LS Kachalova MV, Polyakov SV ³Investigation and development of the intelligent voice assistant for the IOT using machine learning´Moscow workshop on electronic technologies, 2018.