Hindawi
Advances in Human-Computer Interaction
Volume 2023, Article ID 9959015, 17 pages
https://doi.org/10.1155/2023/9959015

Research Article

Afan Oromo Speech-Based Computer Command and Control: An Evaluation with Selected Commands

Kebede Teshite,[1] **Getachew Mamo**,[2] **and Kris Calpotura**[1]

[1]*Faculty of Electrical and Computer Engineering, Jimma University–Institute of Technology, Jimma, Ethiopia*
[2]*Faculty of Computing and Informatics, Jimma University–Institute of Technology, Jimma, Ethiopia*

Correspondence should be addressed to Kebede Teshite; kebe.tesh@gmail.com

Received 19 June 2023; Revised 15 August 2023; Accepted 29 September 2023; Published 16 October 2023

Academic Editor: Shafqat Shad

Speech-based computer command and control utilize natural speech to enable computers to understand human language and execute tasks through commands. However, there has been no study or development of a speech-based command and control system for Microsoft Word in Afan Oromo. The primary aim of this research is to investigate and develop a speech-based command and control system for Afan Oromo using a selected set of command-and-control words from MS Word. To accomplish this objective, a speech recognizer was developed using the HTK toolkit, employing a small vocabulary, isolated words, speaker independence, and HMM-based techniques. The translation of the selected MS command words from English to Afan Oromo was completed in order to develop this automatic speech-based computer command system. Audio recordings were obtained from 38 speakers (16 females and 22 males) aged between 18 and 40 years, based on their availability. Word-level speech recognition was performed using MFCC and data processing, which are widely used and are effective approaches in speech recognition. Out of a total of 64 MS command words, 54 words (84.37%) were used for training and 10 words (15.63%) were used for testing. Live and nonlive evaluation techniques were employed to assess the performance of the recognizer. The live recognizer, which considers variations in the environment, outperformed the nonlive recognizer due to the influence of neighboring phones. The performance results for the monophone tied state, triphone, and triphone-based recognizers were 78.12%, 86.87%, and 88.99%, respectively. Thus, the triphone-based recognizer exhibited the best performance among the nonlive recognizers. The challenges of limited resources in this research study were limited to investigate speech-based commands for computers using only selected MS commands, which play a crucial role in text processing. In order to evaluate a speech-based interface in a real environment, there were no components available for object-as-a-service. The experimental findings of this study demonstrated that if an adequate amount of language resources was available, a computer-based Afan Oromo speech-based interface for command-and-control purposes could be developed.

1. Introduction

Speech is the simplest and most vital form of communication [1]. Through this speech-based Afan Oromo interface, users can communicate with the computer by speaking commands instead of relying on standard input devices such as keyboards and mice. Humans naturally utilize acoustic, lexical, and contextual information to communicate through speech in a naturalistic manner [2]. To achieve computer-generated speech that resembles human speech, it is important to explore systems that can understand human speech without the need for human interpretation [3, 4].

The process of converting speech sound waves into readable text, allowing the computer to hear and respond automatically to human speech, is known as speech recognition [5]. Speech command-based applications are widely used in various fields and have significantly enhanced human-computer interaction [6]. Speech recognition interfaces are integrated into digital devices, e-commerce, e-learning, the Internet of Things, robotics, and medical equipment to facilitate control and monitor through speech input [7, 8].

Speech is a natural, flexible, and simple mode of communication [9]. Therefore, in situations where a computer is

used in the dark, the user's hands are occupied, or speech recognition is needed for a specific purpose or in remote areas where manual input is not feasible; it is advantageous to employ speech-based computer commands for more natural and comfortable communication. Speaking vocal commands directly to applications, instead of relying on a mouse and keyboard to manipulate text, accelerates communication in human-computer interactions. However, achieving accurate automatic speech recognition (ASR) remains a major challenge due to factors such as speaker and language variability, vocabulary size, and noise interference [10, 11].

While speech is a natural, flexible, and effortless mode of communication, the relationship between the physical speech command signal and the corresponding words is exceedingly complex and difficult for computers to comprehend [12]. In the natural world, speech sounds vary across individuals based on factors such as age and gender, which poses challenges for machines to differentiate. The accuracy of an ASR system improves as the vocabulary size increases because the model is trained on a larger dataset. In addition, algorithms and features also play influential roles in speech recognition [13]. Labeling and mapping sequences of speech vectors to symbol sequences require precise boundaries, as words can be spelled in various ways and changes in accents significantly impact speech recognition accuracy [14].

The approach to explore automatic speech-based recognition depends on the amount of collected data and the study's objectives. The development of speech-based ASR has been achieved through algorithms such as hidden Markov model (HMM), recurrent neural network (RNN), deep neural network (DNN), convolutional neural network (CNN), hybrid hidden Markov models with multilayer perceptron (HMM-MLP), and hybrid hidden Markov models with the deep neural network (HMM-DNN) [15–17].

Some people confuse multimodal speech with automatic speech recognition command and control, although the two are fundamentally distinct. Multimodal speech employs various communication channels for human-computer interaction, while speech recognition commands do not [18]. Speech-based computer command and control are used for speech output, whereas a recognition system is used for speech input.

Therefore, the primary goal of scholars investigating speech recognition is to develop a better recognizer that can accurately transform sequences of feature vectors into words using phonetic and linguistic information, enabling human-computer conversations on any topic, in any environment [19].

This research is primarily intended for individuals with special needs. Speech-based computer commands enable users to interact with devices using spoken language, benefiting those with disabilities and providing a hands-free operation. This technology comprehends natural language and finds applications in virtual assistants and dictation, but it faces challenges with accuracy in noisy environments. Ensuring security, expanding language support, and integrating with AI are important considerations. The advancement of this technology enhances user experiences and accessibility while driving innovation in user interfaces. These commands, which utilize natural language and oral speech, offer convenience to people with disabilities such as arm impairment or visual impairment, as well as those with busy hands. This technology has applications in various areas, such as Windows applications, Wi-Fi control, nuclear reactions, medical devices, robots, and electronic/computing devices. The specific focus of the research is the development of speaker-independent speech-based computer commands for the Afan Oromo language. The collected speech data can be utilized for various purposes, including Google voice recognition for Afan Oromo commands and evaluating the effectiveness of Afan Oromo speech-based computer commands.

Furthermore, this should encourage researchers to explore Afan Oromo speech-based computer commands and controls, given the limited work in this area and the growing need for computer commands and controls in the Afan Oromo language. This research aims to examine the feasibility of utilizing speech-based computer commands for controlling commands during text processing.

2. Literature Review

The way humans interact with machines has been transformed by automatic speech recognition (ASR) technology, which has found extensive applications across various domains. These include voice assistants, transcription services, call center automation, and language learning tools. ASR systems play a pivotal role by converting spoken language into written text, facilitating seamless and efficient communication between humans and machines. As the demand for accurate and robust ASR systems continues to increase, it becomes crucial to explore the present state of the field, identify key research trends, and address associated challenges.

The main objective of this literature review is to thoroughly examine the advancements, techniques, and challenges in the realm of ASR. Through a comprehensive analysis of existing research, we aim to gain a deeper understanding of the fundamental principles, methodologies, and algorithms employed in ASR systems. In addition, we will explore the diverse applications of ASR technology and its impact on various industries and sectors.

Recent years have witnessed remarkable progress in ASR technology, owing to advancements in deep learning techniques and the availability of extensive speech datasets. However, despite these advancements, several challenges persist. These include effectively handling variations in speech patterns, mitigating the impact of background noise and environmental conditions, and accommodating the intricacies of diverse languages and accents. By critically analyzing the existing literature, our goal is to identify the current state-of-the-art ASR models and techniques while also highlighting ongoing research efforts aimed at addressing these challenges.

The application enables elderly and disabled users to effortlessly control their electrical appliances within the Internet of Things (IoT) environment without the need for physical movement. By employing speech recognition technology, Apple Siri, Amazon Alexa, and Google Assistant are compared in terms of their effectiveness in executing IoT-based voice commands using machine learning. Through experiments involving smartphones, smart speakers, and control systems, Google Assistant demonstrates the highest pronunciation accuracy at 95%, while Apple Siri exhibits the lowest performance at 80%. However, the study's findings indicate that the development of online and real-time interaction systems for security applications in smart cities and offices will continue to be challenging due to the limitations of IoT technology [8].

To create a speech control interface for computers, a GMM model with a performance rate of 74.38% has been developed. Future research should focus on expanding the repertoire of commands to enhance the efficiency of the interface [20].

Based on empirical data collected from structured user-centered activities involving military personnel, an AI-based decision support for speech command and control system, when successfully implemented, can provide the advantage of faster information analysis, enabling quicker decision-making and operational superiority over adversaries. AI support for execution may involve evaluating action alternatives for commanders and facilitating various types of operations, such as using speech-to-text tools for swift and accurate communication during briefings. Ensuring transparency is a critical challenge in military decision support, where the ability to explain recommendations, understand, and rely on the system is of utmost importance [21].

A voice-controlled electric fan can comprehend spoken commands, offering a more efficient user experience due to its ability to capture commands more rapidly compared to writing or typing. In terms of speed, the system can capture the user's speech at a faster rate than relying solely on the electric fan. According to the results, the Filipino voice commands "IKALAWA" and "IKATLO" exhibit the highest accuracy rate at 100%, while the command "ISA" yields a 50% success rate and "PATAYIN" yields 60%. However, the drawback of this speech-based voice command system is that it only caters to Filipinos and is not universally applicable as a language [22].

This speech interface empowers users to execute common computer commands using the Gaussian mixture model (GMM) and a future technique called Spectral Feature-based Speech Recognition (SFBSR) for speech interface control on PC Windows. The best performance is achieved with 64 centers and 200 iterations. However, as the number of iterations increases, so does the computation time required for creating the GMM. Despite the study using a limited number of speech commands for interface development, the average recognition performance of speech commands reaches 74.38% [20].

In 2003 [23], Martha attempted Amharic speech recognition system for computer command and control based on an experiment with Microsoft Word using the HMM model. A corpus was prepared by participating 10 females and 16 males in the age range of 20 to 35 years. The application operates using a limited set of vocabularies, employing a specialized Amharic word recognizer designed for individuals aged 20 to 35. Training the recognizer involved using 76.9% of the recorded data, while the remaining data were used to assess the recognizer's performance. As a result, the model has performed less (80%) to achieve a goal. The accuracy result of 80% is acceptable, but the researcher used a very small amount of data, and the evaluation matrix for the researcher is only live. In addition, the interface developed for isolated Amharic words cannot be integrated into Microsoft Word even though the researcher did not recommend the service provided for this purpose.

An evaluation of an Amharic speech-based dictation system is examined for the judicial domain with an accuracy of 84.550% and with a word error rate of 16.475%, using an HMM approach (sphinx tool) on continuous speech data for training and building two acoustic models and text data for building a language model. The results demonstrated acceptable performance, but no work in transcription for the indirect court [24].

A spontaneous, speaker-independent Amharic speech recognizer was developed in this research work using training data that consist of 9460 unique words and is approximately 3 hours and 10 minutes of speech. According to this study, the best recognizer performance is 41.60% word accuracy for speakers involved in training, 39.86% for test data from both speakers involved and not involved in training, and 23.25% for speakers not involved in training. The recognizer was developed and tested using less frequent nonspeech events and had lower word accuracy than those that included them [25].

Speech segmentation for Amharic investigation extraction of information from a large archive requires both the extraction of audio file structure and the extraction of speech recognition information. Using a monosyllable acoustic model and forced alignment to segment speech automatically, promising results are obtained. The most accurate results were achieved with a decision tree classifier. The highest evaluation was achieved with segmentation accuracy of 91.93 and 85%, respectively, for reading loud and spontaneous speech. Other evaluation techniques are not applied to the recognizer [26].

Currently, people communicate with electronic devices through their speech with the help of an automatic speech recognition system. A Sphinx 4 tool was used in this study to investigate the feasibility of developing automatic speech recognition for the Ge'ez language using HMMs. A total of 79.70% word accuracy was shown in two experiments conducted online and offline with the Sphinx tool. Test results show that the system using the developed interface has a word accuracy of 67.79%. In this study, recognition accuracy increased when the corpora size was maximized [27].

Afan Oromo language with phonetic recognition and syllable-based recognition HTK tool used to collect speech corpora from 39 males and 24 females of various dialects, with speeches lasting approximately four hours for training

and 40 minutes for testing. Syllable recognition showed promise, but the overall accuracy of this model is low, increasing from 39.55%, 47.21%, 55.35%, and 43.96% with monophones, triphones, tied-state triphones, and syllable-level recognition, respectively [28].

In this study, HTK tool was used to approach HMM modeling techniques for large vocabulary, Afan Oromo speaker independence, and continuous speech. For this study, 2953 utterances (approximately 6 hours of speech) were collected from 57 speakers (42 males and 15 females). Increasing the Gaussian tuning parameters for word insertion penalty to 1.0 and grammar scale factors to 15.0 according to the researchers can improve system performance. Acoustic models that are context-independent (based on monophones) and context-dependent (based on triphones) have been developed. In terms of word error rate, the results for context-independent were 91.46% and 89.84%, respectively. The outcome, however, is best for the study which must be assisted by another research [29].

In 2016, the authors conducted research on the feasibility of developing a large vocabulary speaker-independent continuous speech recognition system for Afan Oromo experimentation, and the bigram language model performed the best with 93% word accuracy for the speaker-dependent test dataset and 43.6% for the speaker-independent test dataset using the HMM approach. When compared to the previous researchers' findings, this one is less accurate [30].

To get a better understanding of ASR and ASR speech-based computer commands, we reviewed various literature sources. A significant change in overall accuracy in speech models is observed because of advancements in open source toolkits HTK, CMU-Sphinx, and Kaldi and their fast-processing speed-based ASR speech recognition. The performance of a speech system is difficult because it is dependent on variations in speakers, their pronunciations, the rate at which they speak, and the dialects of the regions they belong to [31]. ASR speech-based computer command recognizer accuracy varies with ambiguity and vocabulary size; hence, hybrid HMM works best for large vocabulary and HMM works best for small vocabulary [32].

In this preliminary literature review, most studies on speech ASR and speech-based computer commands have been conducted in international languages, with few studies on Afan Oromo and Amharic local languages. The application of Amharic speech commands is related to my field of study, but there is still no Afan Oromo speech-based computer command. This indicates a research gap that motivated me to investigate Afan Oromo speech-based computer commands using HMM model evaluation with selected commands.

3. Materials and Methods

The overall structure of our system consists of three essential components, each playing a vital role in facilitating efficient communication through speech-driven computer commands. These key elements encompass the command text translation module, the Afan Oromo ASR (automatic speech recognition) module, and the communication interface.

Command text translation: The command text translation module functions as a conduit between English and Afan Oromo command texts. Its primary purpose is to take selected command texts in English and accurately convert them into their corresponding Afan Oromo equivalents. This step is crucial to ensure that users comfortable with Afan Oromo can seamlessly interact with our system, thereby enhancing accessibility and usability for a broader audience.

Afan Oromo ASR: At the core of our speech-based interaction system lies the Afan Oromo ASR module. It takes the translated Afan Oromo command texts, coupled with the necessary data for robust speech recognition. To achieve this, we prepare corpora, which comprise various datasets, including training speech data, training text data, the acoustic model of speech units, and a statistical language model. These components work collaboratively, enabling our system to recognize and comprehend spoken Afan Oromo language.

Communication interface: The communication interface serves as the point of interaction between users and our system. It acts as the gateway through which users convey their commands, and our system responds with appropriate actions. Supported by the models generated by the Afan Oromo ASR module, this interface empowers users to communicate their intentions using spoken Afan Oromo commands. It serves as a versatile tool, facilitating hands-free operation and providing benefits to individuals with disabilities, to those with busy hands or those seeking a more natural and convenient way to engage with computers and devices.

Figure 1 shows the general architecture of the proposed Afan Oromo ASR speech-based computer command.

3.1. The HTK Software Toolkit. The HMM model is used to train and decode the recognizer for the application's speech-based computer command. Although various tools including HTK, Sphinx, and Kaldi have been selected and designed for building HMM speech-based audio data processing for particular ASR isolated word-level recognition [33]. HTK, the most popular toolkit for building Hidden Markov models, was created especially for the implementation of speech-based isolated word recognition [31]. Therefore, HTK toolkits were selected for the investigation of Afan Oromo isolated speech-based recognition computer commands. HTK toolkit libraries play a vital role in tasks such as training to estimate a set's parameters HMM transcription of unknown utterances and decoding speech signals [14]. Figure 2 shows many tools used in HTK.

The HShell library module in the HTK program oversees user interactions with an operating system, managing input and output. HLM is utilized for preparing language model files, HNet for creating word networks and lattices, HDict for generating dictionaries, HVQ for forming VQ

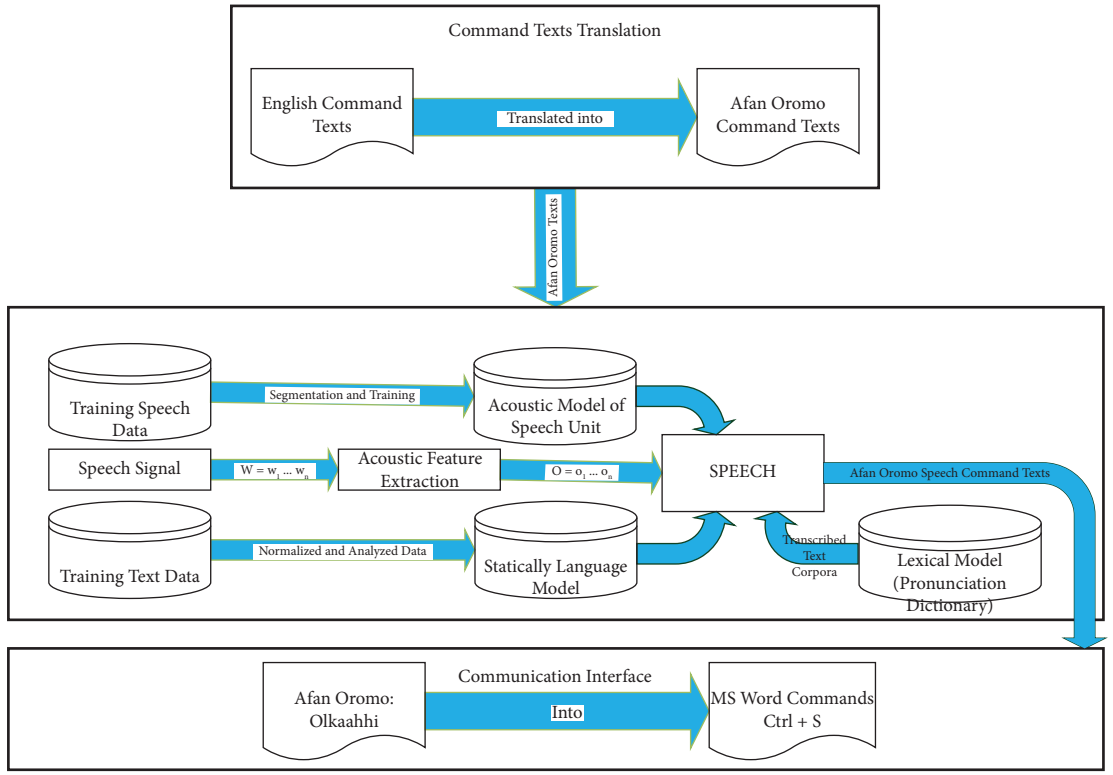

Figure 1: The architecture of AO speech-based computer command and control.

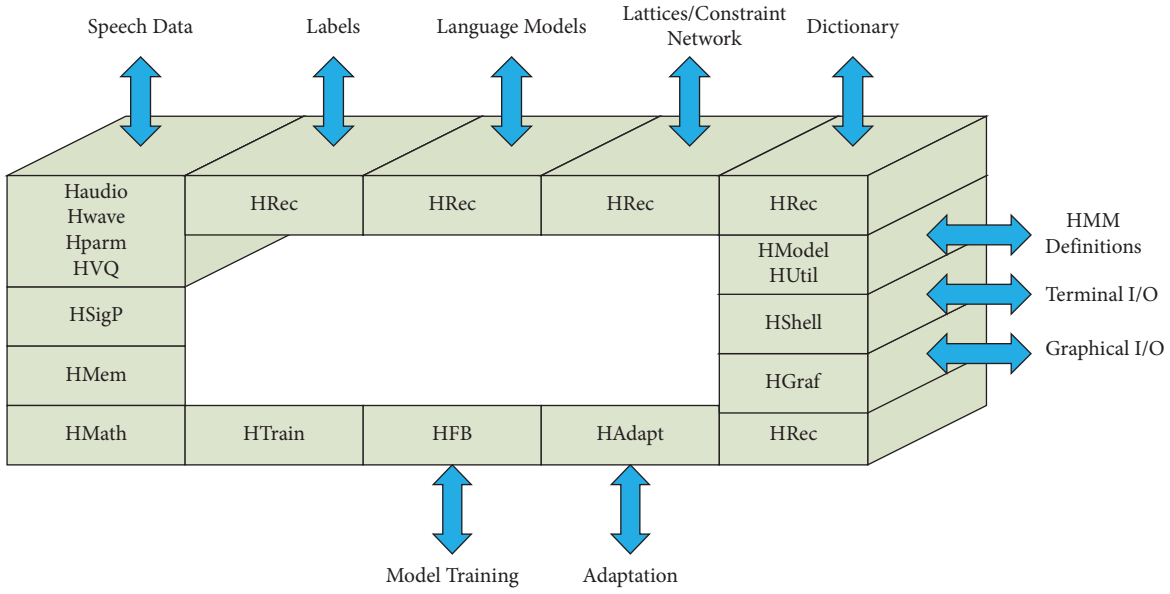

Figure 2: HTK software architecture.

codebooks, and HModel for crafting HMM definitions. HLabel provides an interface for labeling files. Both HWave and HParm handle waveform and parameterized speech input and output, respectively. They support various file types and offer a consistent user interface, facilitating data importation from other systems. Direct audio input is supported through HAudio, and HGraf provides basic interactive visuals. Unlike HTrain and HFB, which include support for various HTK training tools, HUtil incorporates HAdapt to offer a range of utility functions for modifying HMMs, including support for various HTK adaptation tools. Last but not least, HRec encompasses the main recognition processing operations, while HMem manages all memory-related issues. HSigP contains the necessary signal processing procedures for speech analysis, and HMath supports mathematical operations.

3.2. The Proposed Afan Oromo Speech-Based Computer Command Prototype. The development of the prototype involves a systematic approach that can be broken down into three distinct phases: data preparation, recognizer training and testing, and analyzer performance analysis. Each of these phases plays a critical role in creating the proposed Afan Oromo speech-based computer command prototype, as illustrated in Figure 3.

In the initial phase known as data preparation, we embark on the journey by utilizing the translated MS command as a starting point. This phase serves as the foundation for the subsequent stages. We meticulously extract features from these translated commands, which are then stored methodically in the Afan Oromo speech command repository. Simultaneously, the text from these commands is archived in both the Afan Oromo phone dictionary and the Afan Oromo command language model. This comprehensive archive ensures that all relevant linguistic elements are preserved and readily available for further processing.

The Afan Oromo phone dictionary subsequently undergoes a series of crucial transformations, including segmentation and transcriptions. These refined linguistic units are then seamlessly integrated into the acoustic model. The purpose of this integration is to establish a bridge between linguistic representations and acoustic patterns, enhancing the model's ability to interpret and analyze spoken commands effectively. The culmination of this phase involves preparing the data specifically for MFCC (Mel-frequency cepstral coefficient) feature extraction. This meticulous preparation results in the creation of the training, testing, and transcribed text corpus, which serves as a cornerstone for the subsequent phases.

Transitioning to the recognizer training and testing phase, we capitalize on the prepared transcribed text corpus. Here, the spotlight is on our acoustic model enriched with linguistic insights. The process encompasses not only model training but also rigorous testing, including segmentation and transcription tasks. The meticulously curated training corpus is introduced to the acoustic model to further enhance its proficiency. It is noteworthy that this step precedes actual model training, as it is imperative to equip the acoustic model with essential linguistic tools, such as the phone dictionary and the translated MS command, before formal training ensues.

Meanwhile, the test dataset steps onto the stage as a critical player. This dataset undergoes the model's scrutiny, assessing its performance against the diverse set of commands it might encounter. This rigorous evaluation provides valuable insights into the model's strengths and areas for improvement. The evaluation process delves deep into decoding the model's output, considering factors such as segmentation and transcription accuracy. The ultimate goal of this phase is to refine the model's capabilities, ensuring that it accurately comprehends and responds to spoken commands.

The culmination of these efforts is manifested through the integration of the acoustic model with the analyzer performance analysis. This integration is facilitated via a communication interface that promotes seamless interaction between the two components. The interface acts as a conduit for sharing crucial information between the model and the analyzer, enabling a comprehensive assessment of the prototype's performance. Notably, the interface relies on data derived from our MFCC feature extraction, establishing a strong link between the preparatory phases and the final analysis.

Throughout the entire journey encompassing each phase, a diverse array of HTK tools is judiciously employed. These tools contribute to the refinement, enhancement, and analysis of the prototype, ensuring that each step is meticulously executed with precision.

The proposed Afan Oromo speech-based computer command prototype embodies these three interconnected phases. From data preparation's meticulous groundwork to recognizer training and testing's model refinement culminating in analyzer performance analysis's comprehensive evaluation, every aspect of the prototype's development is a testament to the meticulous planning and execution that underpins its creation.

3.3. Data Preparation. The initial stages of preprocessing computer text commands involve the translation and transcription of these commands, as well as the creation of phone-based dictionaries. Transcription occurs at two levels: the word level and the phone level. For the transcription at the phone level, we utilize a pronunciation dictionary file containing a list of words and their possible equivalent phone sequences.

The transcribed files, along with the acoustic features of the speech signals, are essential for the development of a recognition model. To obtain audio signals for this process, different speakers read the translated script of each selected text computer command. The preparation includes noise reduction using both front-end and back-end methods. This prepared data are then used to train and test the Afan Oromo speech-based recognizer, employing the HTK toolkit. This comprehensive approach ensures that our recognition model is effectively trained and evaluated.

3.3.1. Command Translation. In this study, our initial step is to select an English computer command and translate it into an equivalent command in the Afan Oromo text corpus. We are seeking the expertise of a linguistic professional to assist us in this process. The commands chosen for translation are Microsoft shortcut commands specifically designed for word text processing. Such commands are Copy, Cut, Paste, Save, Open, and so on.

3.3.2. Pronunciation Dictionary. To train and transcribe at the phones' level, language lexicons or pronunciation dictionaries are essential files. The researcher developed Python code to generate phone-based pronunciations for each word. The prepared pronunciation dictionaries include both phone-based and alternative versions. It is important to note that, for this research, the specific pronunciation dictionary

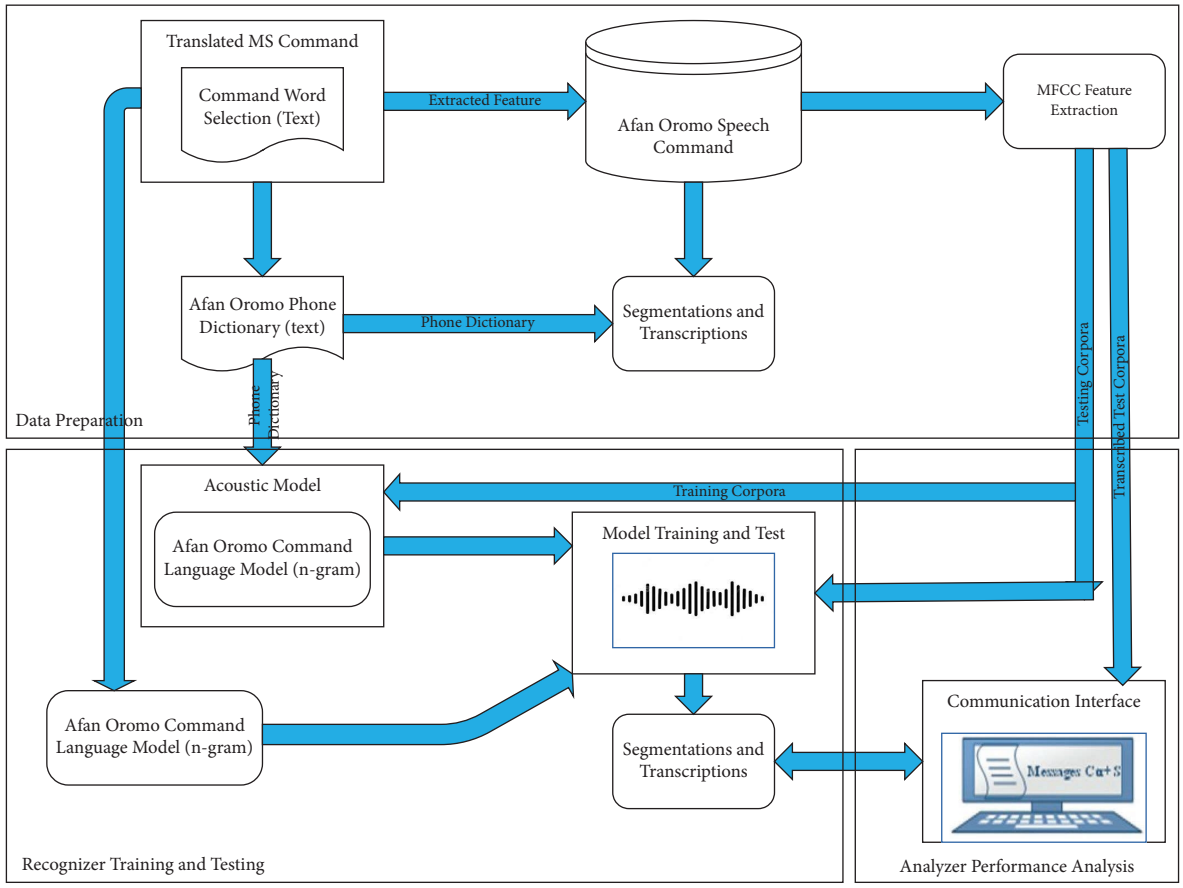

Figure 3: Proposed Afan Oromo speech-based computer command prototype.

has not yet been created or published. For this study, the researcher had the task of preparing pronunciations that would be utilized. The initial step involves creating a phone-based dictionary for training and labeling purposes, utilizing recorded computer command words from the text corpus. This begins by generating a list of words extracted from the text corpus. A Perl command is then executed through a Perl program to facilitate the creation of these wordlists.

3.3.3. Transcribing Segmented Speech. To create a speech corpus, our initial step involves transcribing the chosen text computer command, which serves as the foundation for the speech corpus preparation. We carefully select the source and define the scope of the text to be used. Transcribing segmented speech is essential for constructing an acoustic model. In this process, we utilize the Afan Oromo alphabet, known as "Qubee," to represent phone numbers in each segment. Following the Afan Oromo writing rules, we construct words.

It is important to note that some phonemes in the IPA representation of the Afan Oromo alphabet cannot be represented in ASCII and are consequently unsupported by HTK (a tool or software). To handle the transcription of these letters, we apply the same method, except for glottal sounds, which are treated differently. For instance, "c" corresponds to "c," "ch" represents "ch," and so on. However, for glottal sounds, we use "hh." In Afan Oromo, the "h" phoneme is not duplicated. You can find the general IPA phonemes used in the transcription, along with their IPA equivalents in Table 1.

3.3.4. Speech Data Collection. In this research investigation and analysis of computer commands conveyed through speech, the absence of pre-existing speech datasets necessitated the creation of a new corpus from the ground up. This process was notably time-intensive. The speech dataset was exclusively compiled from specific text commands used in Microsoft applications, which are vital for text-based operations. To capture the audio signals of spoken commands, translated text-based computer instructions were sourced from multiple speakers. Among these speakers, a subset was selected and requested to articulate the provided scripts from the text corpus. The report lacks information regarding the distribution of male and female speakers or their age ranges. The determining factor for selection was speaker availability. The recording sessions took place in both a high school environment and the Oromia Science and Technology video conference room, chosen to minimize background noise interference.

In total, 38 speakers participated in this study, ranging in age from 18 to 40 years. Out of these, 32 speakers (18 males and 14 females) were employed for training, constituting

Table 1: Phoneme with corresponding IPA.

Letter	A	AA	B	BB	C	CC	CH	D	DD	DH	E	EE	F	FF
IPA	A	Aa	b	bb	C	Cc	Ch	D	dd	dh	e	Ee	f	ff
Letter	G	GG	H	I	II	J	JJ	K	KK	L	LL	M	MM	N
IPA	G	Gg	h	I	Ii	J	Jj	K	kk	L	ll	M	mm	N
Letter	NN	NY	O	OO	P	PP	PH	Q	QQ	R	RR	S	ss	sh
IPA	Nn	Ny	o	oo	P	Pp	Ph	Q	qq	R	rr	S	ss	sh
Letter	T	TT	TS	U	UU	V	W	WW	X	XX	Y	YY	Z	‘
IPA	T	Tt	ts	U	Uu	V	W	ww	x	xx	y	yy	z	hh

84.3% of the overall speaker pool. In addition, 6 speakers (three males and three females), making up 15.7% of the total, were reserved for testing purposes. This specific age bracket (18–40 years) actively utilizes Microsoft Word for text manipulation. Subsequent to amassing the speech data, it was divided into distinct sets for training and testing. The test set served to assess the performance of the recognition system, while the training set was instrumental in training the recognition model. Prior to collecting the speech data, a comprehensive speech text corpus was meticulously constructed. The English commands commonly used in Microsoft Word were painstakingly translated into prompts in the Afan Oromo language. The collected speech data underwent recording, segmentation, coding, and parameterization processes to extract its acoustic characteristics. The Mel-frequency cepstral coefficients (MFCC) technique was employed to simultaneously extract all relevant acoustic features. The HCopy tool was employed to extract acoustic information from the recorded utterances. The parameterization of speech data could be done in real time or collectively, extracting all parameters using the HTK tool.

3.4. Data Preparation Phase. The speech corpus was collected and then divided into training and testing sets in order to train the recognizer. The training set was used to train the recognizer, while the test set was used to evaluate its performance. Multiple experiments were conducted to explore a functional prototype of a command-and-control system for an Afan Oromo speech-based computer. This chapter provides a description of the key ASR tasks and components used in the statistical approach, including data preparation, recognizer training, recognizer testing, and analyzer performance analysis. Figure 3 illustrates the developed system architecture and the results obtained from the HTK tools. The development of speech recognition systems involves the crucial step of data preparation, which is undertaken by numerous researchers during the development process. To construct HMM speech-based models, it is essential to possess a collection of speech data files containing transcriptions and translations. Within the data preparation phase, various steps can be undertaken, such as creating pronunciation dictionaries, transcription files, and coded audio files.

3.4.1. The Task Grammar. In the process of developing a prototype recognizer, a word-level network called task grammar is used to define valid word sequences. The grammar takes the form of a simple job grammar, specifying a structure of three words in the sequence: silence, any command word, and silence again. This task grammar is created using the HTK grammar definition format, specifically the HParse format resembling EBNF (extended Backus–Naur form), where choices are denoted by vertical bars. To convert the HParse format into a standard lattice format (SLF) word network, a conversion is required in HTK version 3.4, the version used in the experiment. The conversion is done using the HTK program HParse, which automatically translates the HParse format to SLF. The resulting SLF is then used in the HTK recognition tool. Figure 4 presents a visual representation (network) of the task grammar.

3.4.2. Command Word Selection. Despite its limitations, Microsoft Word commands play a vital role in text processing. It is crucial to understand that every action in Microsoft Word is achieved through commands. For instance, the file menu contains a greater number of commonly used commands compared to other menus. Commands such as table (GAbATEe), open (bANI), save (OLKAaI), and print (MAXxANSI) represent specific actions within Microsoft Word. Each command has its own distinct function, allowing for precise and detailed tasks to be performed in Microsoft Word.

3.4.3. Pronunciation Dictionary. The process of creating a pronunciation dictionary for HTK comprises multiple steps. Initially, a wordlist is generated from the training transcription, which includes unique words. Through the use of a Perl script called prompts2wordlist, separate wordlists are created for training and testing purposes. These wordlists are then utilized to compile a sorted wordlist for the dictionary. In this particular research, a manual pronunciation dictionary named dict_phones.lex is developed, encompassing 56 Afan Oromo phones. These phones are derived from a combination of 40 consonants, 6 compound symbols, and 10 vowels. The dictionary is standardized by utilizing IPA symbols; however, some manual corrections are necessary due to the limitations of ASCII code. Table 1 offers an illustrative representation of the employed phones. The HDMan command is employed to search for word pronunciations within the source dictionary (dict_phone.lex) and output the results in a dict_phones.lex file. This process involves the creation of monophone lists, removal of specific phones, and the generation of a new dictionary

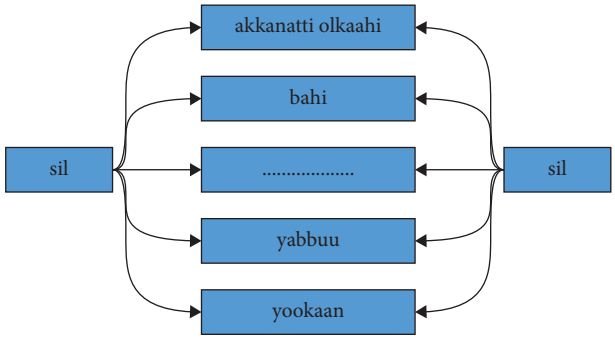

Figure 4: The word network.

called "dicts." Each entry in the dictionary includes the word and its corresponding phonetic sequence. The training vocabulary consists of 56 Afan Oromo phones, including symbols such as "sp" and "sil."

3.4.4. Creating File Transcription. To prepare speech data for training, file transcriptions need to be created with proper formatting and assigned labels. Two types of transcriptions can be used: word-level and phone-level. For word-level transcriptions, an orthographic transcription in HTK label format is required. This can be achieved by either creating separate label files for each line of the Word file or using a programming language to construct a master label file (MLF). In this experiment, the second method was chosen, and a Perl script called prompts2mlf was used to generate the .mlf file, resulting in a file named trainwords.mlf containing word-level transcriptions. To convert the word-level transcriptions to phone-level transcriptions, the HLEd command from the HTK tool was used. This command substitutes each word with its corresponding phoneme by looking up the phones in the prepared dictionary file. The output is stored in a file called phones0.mlf, which does not include short pauses (sp) after each word-phone group. The HLEd command is executed with the mkphones0.led modifying script. Afterward, the HLEd command is run again with the mkphones1.led editing script to create a phones1.mlf file that includes short pauses (sp) after each word-phone group. Overall, the process involves creating the trainwords.mlf file for word-level transcriptions followed by generating the phones0.mlf and phones1.mlf files for phone-level transcriptions using the HLEd command with specific modifying and editing scripts.

3.4.5. Back-End Feature Extraction. The back-end feature extraction process involves converting speech waveforms into parameter vectors. The HTK tools offer on-the-fly coding from the original waveform file, which requires additional preprocessing during training. The coding task is crucial in the data preparation stage. To achieve this in HTK, the HCopy tool is used with a configuration file and a script file containing a list of files. The HLEd tool is employed to generate a single MLF file by setting the TARGETKIND configuration variable to MFCC0D. Adding time derivatives to the static parameters improves speech recognition system performance. Since HTK is not efficient in processing.wav files, they are converted to MFCC format using the HCopy program. In the experiment, a file listing the source audio files and the corresponding converted MFCC files are created. The HCopy command is run using this script file as a parameter to extract speech features and generate MFCC files for each utterance. Two script files, codetrain.scp and codetest.scp, are written for training and testing, respectively. The HCopy command is executed with the provided configuration file (HCopy_train.txt) to convert the wave format to MFCC. The conversion parameters are specified in the configuration file. By running the HCopy command, a series of MFCC files is produced corresponding to the audio files listed in codetrain.scp.

3.5. Training Model Selection. Hidden Markov models (HMMs) are generative models based on the Markovian assumption that the current state (S') depends solely on the previous state. HMMs are simpler compared to recurrent neural networks (RNNs) and rely on strong assumptions that may not always hold true. Therefore, for recognizing speech-based computer commands, we require a recognizer that does not necessarily rely on these strong assumptions. RNNs, which are a type of artificial neural networks (ANNs) commonly used for modeling sequential data such as speech signals have the tendency to identify false patterns in the data and overfit. On the other hand, HMMs are a suitable choice when working with a small and simple dataset. In our case, out of the total of 64 Microsoft command words, 54 words (84.37%) were used for training and 10 words (15.63%) were used for testing.

RNNs, however, are well-suited for large-scale datasets. They are designed for complex sequential prediction tasks that involve generating irregular and error-prone sequences as outputs. In speech recognition, when the dataset is small and uncomplicated, the HMM model tends to outperform the RNN model. For our research on Afan Oromo speech-based computer command and control on Microsoft Word, our available data are limited and not complex. Hence, HMMs are considered a more suitable candidate model for investigation compared to RNNs. This is the stage where recognition and decoding occur, utilizing the HMM model, as speech is characterized by its temporal structure and encoded as spectral vectors [34].

3.6. Training Prototype Model. Training prototype models constitutes a pivotal aspect in various fields, including machine learning, computer vision, and speech processing. These models serve as foundational representations capturing essential characteristics of a given dataset. By training on a diverse set of examples, prototype models learn to identify common patterns, features, or behaviors within the data. This enables them to generalize and make predictions on new, unseen data instances. The training process involves fine-tuning model parameters using optimization techniques, enhancing the model's ability to classify, recognize, or generate new data. Prototype models find applications in numerous domains, ranging from

image classification and object detection to natural language understanding, playing a central role in advancing the capabilities of AI systems.

3.6.1. Creating Prototype Monophone. The process of creating a prototype monophone is a crucial phase in HMM model training. The focus of this phase is on specifying the model topology rather than the parameters. In our phone-based recognition system, a 5-state left-to-right HMM architecture is used, consisting of 3 emitting states and 2 nonemitting states (see Figure 5).

The HTK tool HCompV is utilized to calculate the global mean and variance from a set of data files. It sets all Gaussians in a specific HMM to have the same mean and variance. The HCompV command, with appropriate parameters and configuration file (HCopy_proto.txt), is executed using the train.scp file that contains the list of training files. This command modifies the prototype file, replacing zero means and unit variances with global speech means and variances.

With the newly created prototype model from HCompV, a master macro file (MMF) named hmmdefs is manually generated. This file copies the prototype and replaces it for each required monophone, including "sil." The format of an MMF is similar to that of an MLF, eliminating the need for multiple HMM specification files. The macros section of the hmmdefs file includes a global options macro and the variance floor macro (vFloors) previously developed by HCompV. The global options macro defines the HMM parameter kind and vector size.

3.6.2. Re-Estimating Monophones. The HERest program is used to re-estimate the monophones. Multiple iterations of HERest are performed, each with different model directories (hmm0, hmm1, and hmm2) and output directories (hmm1, hmm2, and hmm3). The re-estimation is done using the data listed in the train.scp file and the labels/phones0.mlf file, which contains phone-level transcriptions. Pruning thresholds are set using the -t option to restricting the range of state alignments included in the training process. The thresholds are initially set at 250.0 and increased if re-estimation fails for a file. The updated model set is stored in the hmm directories (hmm1, hmm2, and hmm3) after each iteration. Figure 6 shows the flow of prototype HMM definition.

3.6.3. Fixing the Silence. A 3-state left-to-right HMM is created for each phone, including a silent model called "sil." Additional transitions are added to the silent model, allowing individual states to absorb impulsive sounds in the training data and improve the model's robustness. A single-state short-pause model ("sp") is developed and connected to the central state of the silent model. Figure 7 shows the topology for two silent models.

3.6.4. Realigning the Training Data. The HVite program is used to realign the training data. It takes word-level transcriptions, phone models, and a dictionary as input. The output is a new phone-level transcription file (aligned.mlf) that matches the acoustic data more accurately. The -o SW option is used to include time-stamp information in the alignment output to detect significant stops at the start and finish of utterances. The HMM set parameters are re-estimated using HERest after the new phone alignments have been established.

3.7. Training Prototype Triphone and Tied-State. The training of prototype triphone and tied-state models is a fundamental aspect of acoustic modeling in the field of speech recognition. These models play a crucial role in converting spoken language into text by capturing the acoustic characteristics of speech signals. Prototype triphone models represent phonetic transitions within speech, while tied-state models group similar acoustic states together. This process involves building a comprehensive dataset, extracting acoustic features, and employing techniques such as hidden Markov models (HMMs) to train these models. The goal is to enhance the accuracy and efficiency of automatic speech recognition systems, enabling them to accurately transcribe spoken language into text form. The process of training prototype triphones and tied-state triphones includes the following:

3.7.1. Tied-State Triphones. The first step is to determine whether to use crossword triphones. If so, monophones are converted to triphones, and word boundaries are marked in the training data. Triphone models are developed and re-estimated, and acoustic states are tied to ensure the use of the same parameters. The HERest tool is used to update the context-dependent models.

3.7.2. Making Triphones from Monophones. Monophones are used to create triphones. The HLEd tool is used to create a list of triphones based on the monophone transcriptions. The triphone transcriptions are created by modifying the monophone transcriptions according to predefined rules.

3.7.3. Creating Tied-State Triphones. Once a set of triphone HMMs is prepared, the states of triphone sets are tied to share data and produce reliable parameter estimations. The HHEd tool is used, and two methods are described: one based on data and the other using decision trees. The decision tree searches for contexts that distinguish clusters based on acoustic properties. The tied-state triphone models are updated using HERest.

3.8. Performance Evaluation Technique. The hidden Markov model (HMM) was utilized to develop the speech recognizer, as HMMs are integral to most modern speech recognition systems, particularly those employing statistical methods. The speed of the recognizer is measured in real-time factors, while accuracy is assessed in terms of performance accuracy, typically represented by the word error rate (WER) [35]. The performance of the speech-based

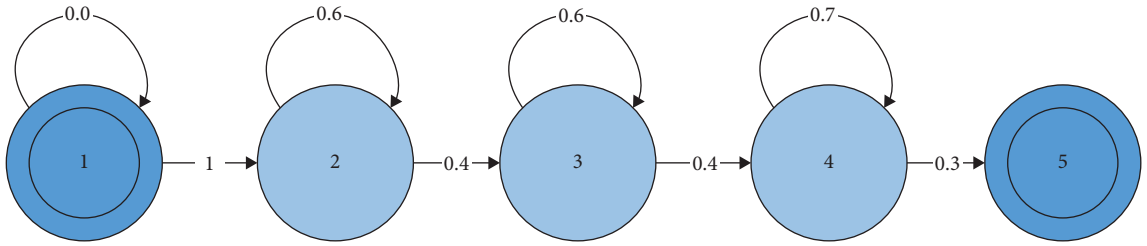

Figure 5: The HMM with 3 emitting states and with skip.

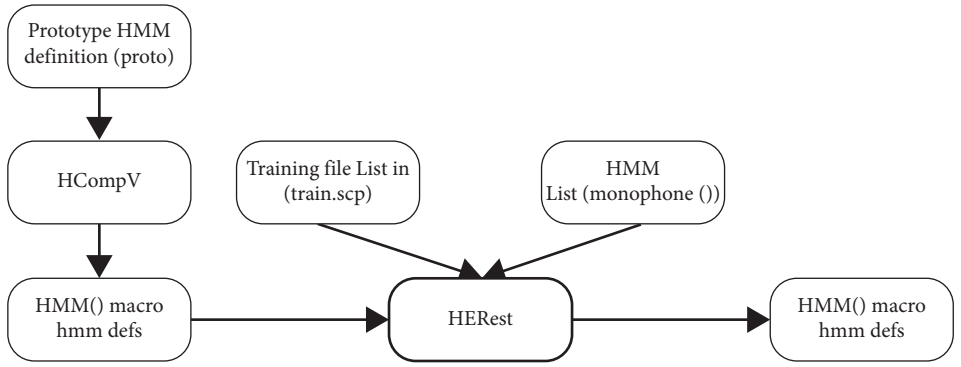

Figure 6: The flow of prototype HMM definition.

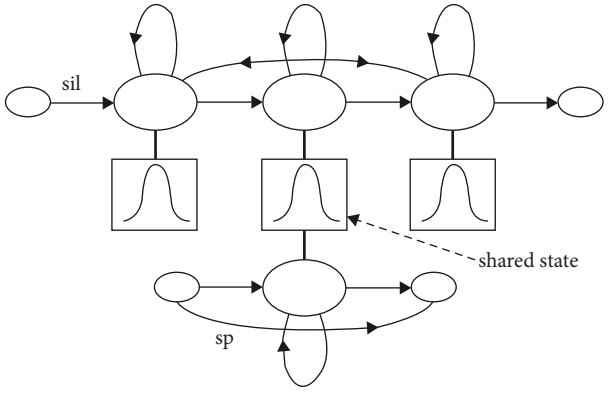

Figure 7: Topology for two silent models.

recognizer is evaluated by measuring the WER and the word recognition rate [36]. Word errors can include insertions, replacements, and deletions. The HResult tool in HTK is employed to analyze the system's performance, comparing the original reference transcription file with the output transcription file generated by the HVite tool. HMMs are recognized as the most powerful statistical tool in automatic speech recognition (ASR) for modeling nonlinearly aligned speech and estimating model parameters [37, 38].

For evaluating the performance of the recognizer, out of a total of 64 Microsoft (MS) commands, testing and training are conducted. The performance of a speech-based command interface can be evaluated using the following equations, where N represents the number of words in the test set, D denotes the number of deletions, S represents the number of substitutions, H stands for word correct unit, and I represents the number of insertions. The internal WER(R), accurate word, and correct word are calculated based on the following formula [35]:

$$\begin{aligned}
\text{WER}\ (\%) &= \frac{\text{Insertion}\ (I) + \text{Substitution}\ (S) + \text{Deletion}\ (D)}{\text{No. of Reference Words}\ (N)} * 100,\\
\text{Correct}\ (\%) &= \frac{\text{No.of} - \text{Ref}\ (N) - \text{Substitution}\ (S) - \text{Deletion}\ (D)}{\text{No. of Reference Words}\ (N)} * 100,\\
\text{Accurate}\ (\%) &= \frac{\text{No. of Reference Words}\ (N) - \text{Substitution}\ (S)}{\text{No. of Reference Words}\ (N)} * 100.
\end{aligned} \tag{1}$$

4. Result and Discussion

The creation of a functional prototype for an Afan Oromo speech-based command-and-control system involves a series of foundational steps, each contributing to its effectiveness. This section delves into these stages, detailing the methodologies employed and their pivotal role in shaping the final outcome. Beginning with the assembly of an extensive speech

corpus representing diverse Afan Oromo commands, the training and testing sets are meticulously divided. The training set enhances the recognizer's ability to understand intricate language patterns, adapting its algorithms to Afan Oromo speech intricacies. Conversely, the test set evaluates the recognizer's proficiency with new commands. This section also highlights the importance of refining the system's performance through experimentation, fine-tuning parameters, and algorithms.

Detailed exploration of the command-and-control system's components is provided, from data preparation to recognizer training, testing, and performance analysis. The holistic approach ensures that linguistic data are meticulously processed, improving the recognizer's learning and evaluation across various scenarios. This understanding is pivotal, serving as a roadmap for developing a seamless speech-based command system where language nuances meet cutting-edge technology. The interconnectedness of data, training, testing, and analysis forms the bedrock of this innovative approach, paving the way for a future where spoken language seamlessly interfaces with digital systems.

4.1. Performance Evaluation. Performance evaluation is a critical process used to assess the effectiveness, efficiency, and quality of a system, process, or entity. It involves systematically analyzing and measuring various metrics to gauge how well the subject performs its intended functions or achieves its goals. Performance evaluation provides valuable insights that help stakeholders understand strengths, weaknesses, and areas for improvement. Whether applied to technology, business processes, or individuals, performance evaluation plays a pivotal role in making informed decisions, optimizing outcomes, and driving continuous enhancement.

To analyze recognizer performance, HTK offers the HResults tool. Test data were fed into the recognizers, and the recognized transcriptions were saved in a separate MLF. HResults were executed with this MLF, which had been created in the data preparation step, to evaluate the performance of isolated Afan Oromo word recognizers. During prototype development, researchers primarily engage in testing and evaluation. Prototypes are evaluated using the most commonly used evaluation techniques: live and nonlive (phone-based). This process is commonly referred to as decoding or recognizing the speech signal. Each word is then represented by hidden Markov models (HMMs) that correspond to the word's sequence of sound units. As a result, the search graph becomes a complex HMM, and recognition is carried out by aligning the search graph with the speech features extracted from the utterance using the Viterbi algorithm.

4.1.1. Phone-Based Recognizer Evaluation. Word-internal triphones, crossword triphones, and tied-state triphones are all considered in phone-based modeling. HVite utilizes phone sets, test data, and output files from n-gram language models, along with a phone-based pronunciation dictionary and other inputs, for the phone-based decoding process. We have created four phone-based systems; thus, the phone sets used for the recognition procedure differ. These phone set lists are grouped into categories such as crossword, monophone, triphone, and tied lists. The text files list the monophone, triphone, and tied-state phone sets, respectively. All phone-based systems employ the same language model, test dataset, and pronunciation dictionary. The word-level transcriptions of each test file are used as the test data. To decode the phone-based systems, you will need to execute the following commands:

For monophones:

HVite -H hmm9/macros -H hmm9/hmmdefs -C configs/hcompv_config.txt -S test.scp −1* -i recog/monorecou.mlf -w wdnet -p 1.0 -s 15.0 dicts/dicts corpus/monophones1

For word internal triphones:

HVite -H hmm12/macros -H hmm12/hmmdefs -C configs/hcompv_config.txt -S test.scp −1* -i recog/wirecout.mlf -w wdnet -p 1.0 -s 15.0 dicts/dicts triphones1

For tied-state phones:

HVite -H hmm15/macros -H hmm15/hmmdefs -C configs/hcompv_config.txt -S test.scp −1* -i recog/witiedrecout.mlf -w wdnet -p 1.0 -s 15.0 dicts/dicts tiedlist

The options -p and -s in this command indicate the word insertion penalty and the grammar scale factor, respectively. A fixed amount is added to each token when transitioning from the end of one word to the beginning of the next, known as the word insertion penalty. The language model probability is scaled before being added to each token during this transition, which is called the grammar scale factor. Given their potential impact on recognition performance, it is highly recommended to tune these factors using development test data.

The following commands are used to assess the recognition results after they have been completed, and the table below displays the results for this triphone model. Comparison of internal words with neighbors in the phone-based recognizer is shown in Tables 2 and 3.

HResults -I labels/testref.mlf corpus/monophones1 recog/monorecou.mlf

HResults -I labels/testref.mlf triphones1 recog/wirecout.mlf

HResults -I labels/testref.mlf tiedlist recog/witiedrecout.mlf

Word-internal triphones, crossword triphones, and tied-state triphones are important considerations in phone-based modeling. HVite, a decoding tool, utilizes phone sets, test data, an output file from the n-gram language model, a phone-based pronunciation dictionary, and other inputs for the phone-based decoding process. Four phone-based systems were created, each using different phone sets for recognition. These phone sets are categorized as crossword, monophone, triphone, and tied lists. The systems employ the same language model, test dataset, and pronunciation

Table 2: Phone-based models' recognition.

Parameter tuning values		Percent of Word monophone			Percent of Word internal triphones			Percent of Word tied-state tri-phones recognized		
-p options	-s options	Correctly recognized	Accurately recognized	Word error rate (%WER)	Correctly recognized	Accurately recognized	Word error rate (%WER)	Correctly recognized	Accurately recognized	Word error rate (%WER)
0.0	3.0	83.02	78.12	21.88	91.51	88.99	11	90.32	86.87	13.12
0.2	5.0	83.02	78.12	21.88	91.51	88.99	11	90.32	86.87	13.12
0.4	7.0	83.02	78.12	21.88	91.51	88.99	11	90.32	86.87	13.12
0.6	9.0	83.02	77.98	22	91.51	88.99	11	90.32	86.87	13.12
0.8	11.0	83.02	77.98	22	91.51	88.99	11	90.32	86.87	13.12
1.0	13.0	83.02	77.98	22	91.51	88.99	11	90.32	86.87	13.12
1.2	15.0	83.02	77.85	22.1	91.51	88.99	11	90.32	86.87	13.12

Table 3: Recognizer variable variance.

	Users involved in training	Total number of recorded MS Word command (54)	Accurate word recognized (%)
1	Trkabada	53	98.14
2	Trnagasuu	54	100
3	Trzerhun	52	96.29
4	#Trrebira	52	96.29
5	Trdeguu	53	98.14
6	#Trayyaantuu	51	94.44
7	Trmooneet	52	96.29
8	#Trdarartu	50	92.59
	Avg	**52.12**	**96.52**

The average number of words for users involved in training is 52.12. Users with a '#' symbol before their name did not participate in live training. Additionally, the average accuracy for word recognition among the 8 users involved in training is 96.52%.

dictionary. The test data consists of word-level transcriptions from each test file. Table 2 shows the phone-based models' recognition.

Table 2 demonstrates that the tied-state and triphone-state models exhibit slightly higher accuracy. The options -p and -s in these commands are used to specify the word insertion penalty and the grammar scale factor, respectively. Word insertion penalties are applied when tokens transition from one word to another. Despite varying the parameters -p and -s, the recognizer's performance remained unaffected.

To assess the accuracy of nonlive recognizers, 38 speakers (17 females and 21 males), aged between 18 and 40, were evaluated based on their availability. Out of a total of 64 MS command words, 54 words (84.37%) were used for training and 10 words (15.63%) were reserved for testing. The monophone tied-state, triphone, and triphone recognizers achieved word-level accuracies of 78.12%, 86.87%, and 88.99%, respectively. Consequently, the triphone-based recognizer outperforms in nonlive recognition performance.

4.1.2. Live Recognizer Evaluation. In general, the process of live speech recognition involves assessing the accuracy of commonly used words and phrases. Due to time limitations and the extended duration of this test, each participant was assigned only eight randomly selected command words to orally control Microsoft Word. To evaluate performance, each person was tasked with commanding and operating Microsoft Word using the randomly selected phrases. Among the eight participants in this study, three were females and five were males.

For recognizer evaluation, the following command was utilized:

HVite -H hmm15/macros -H hmm15/hmmdefs -C configs/hvitelive.txt -w wdnet -p 0.0 -s 5.0 dicts/dicts tiedlist

Table 3 represents the participants who did not undergo training indicated by the "#" symbol. The evaluation of the Afan Oromo speech-based recognizer performance is based on the participants who underwent training and those who did not. For participants not involved in training, the recognizer performance shows a maximum accuracy of 96.29% in Table 3. On the other hand, participants who underwent training achieved a maximum accuracy of 100% in recognizer performance. To calculate the average number of accurately recognized words, the number of correctly recognized words was divided by the total number of available word commands during the live evaluation of the fixed variation.

The performance of the recognizer's variable variance is evaluated by considering the users who participate in training and those who do not. The results presented in Table 4 demonstrate the performance of the users who participated in training, achieving a maximum accuracy of 97.95%. On the other hand, the recognizer's accuracy for users who did not participate in training reached a maximum of 91.83%.

4.2. Prototype Communication. The service-oriented component offers an extension that facilitates software communication with other software, networking with networks,

Table 4: Recognizer in fixed variance.

	Users involved in training	Total number of selected words (49)	Accurate word recognized (%)
1	Rkabada	47	95.91
2	Trnagasuu	46	93.87
3	#Trzerhun	45	91.83
4	Trrebira	47	95.91
5	Trdeguu	46	93.87
6	#Trayyaantuu	40	81.63
7	Trmooneet	48	97.95
8	#Trdarartu	39	79.59
	Avg	**44.75**	**91.32**

The average number of words for users involved in training is 44.75. Similarly, users with a '#' symbol before their name did not participate in live training. Additionally, the average accuracy for word recognition among the 8 users involved in this training is 91.32%.

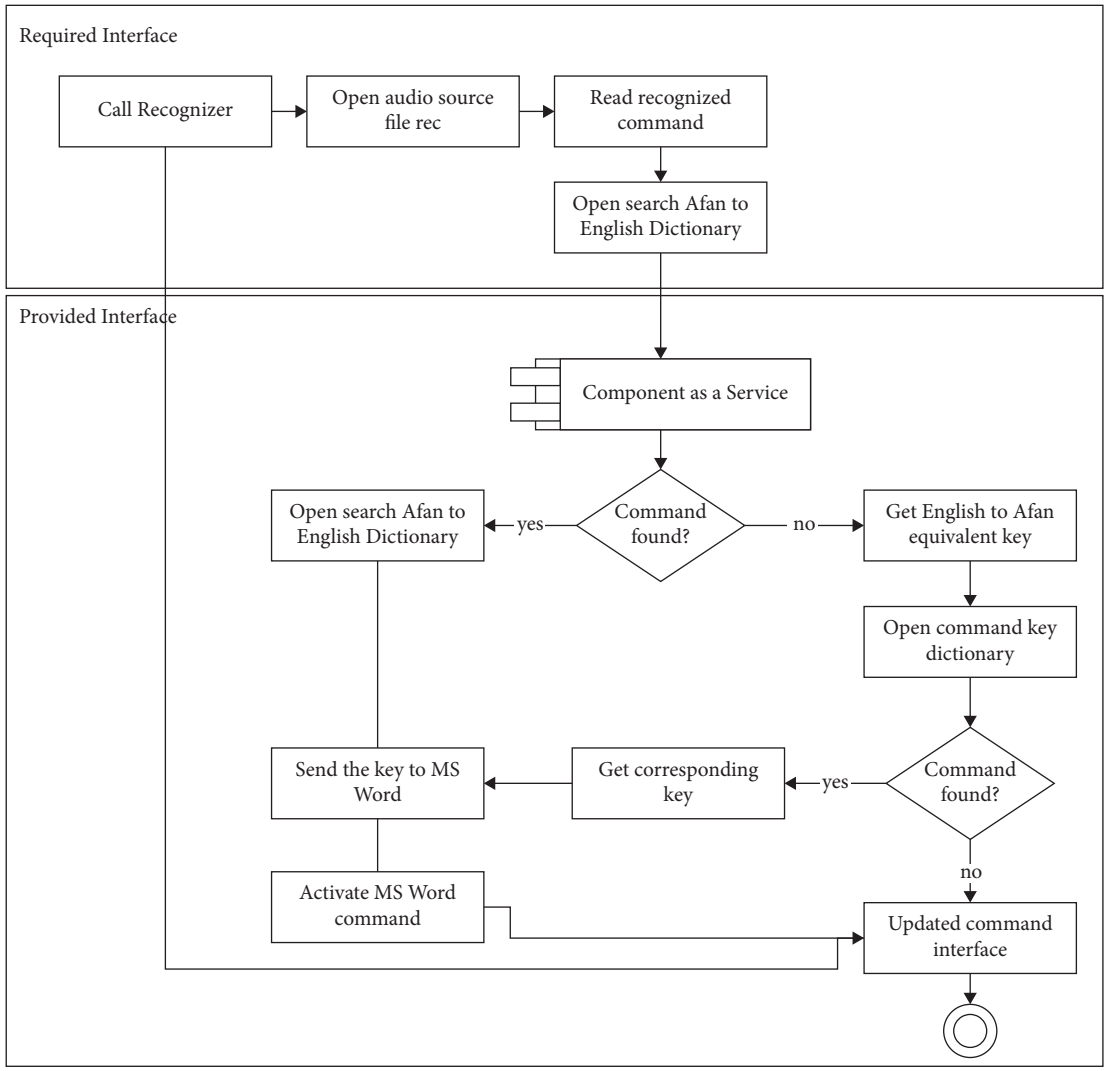

Figure 8: Communication interface flow diagram [39].

and system integration. As shown in Figure 8, a prototype recognizer is developed, and a service-oriented component is designed to enable communication between the recognizer and Microsoft Word. To establish a connection between the recognizer and Microsoft Office, Microsoft needs to authorize the recognizer as a service object. There are two fundamental structures for speech-based computer commands and controls: the provided service object and the required interface, which is the recognizer.

Prior to speech recognition, the recognizer, as depicted in the diagram (see Figure 8), initiates an audio signal recording and proceeds to search for a matching command

key. If the audio matches the command key, the recognizer can execute the corresponding operation by either opening or closing the words. The Afan Oromo speech-based computer command interface utilizes a dictionary to search for Afan Oromo-to-English key commands, enabling communication with the MS computer command. Any updates to the interface require a process of returning, searching, and calling Afan Oromo commands.

4.3. Recognition and Discussion. Decoding algorithms are utilized to accomplish the process of recognition. The objective of this experiment, as mentioned earlier, was to develop a speech interface system that allows users to control the computer through spoken commands. In order to train the Afan Oromo speech-based recognizer, a dataset is prepared by performing translation, transcription, audio data segmentation, and MFCC feature extraction.

To evaluate the performance of the system, both live and nonlive evaluation techniques are employed. In the live setting, the recognizer's performance for Afan Oromo speech-based computer commands is assessed using a fixed variance model (49 words) and a variable variance model (54 words), achieving a maximum accuracy of 91.83% and 96.29%, respectively, for those not involved in training. The fixed variance model is more affected by noise, whereas the variable variance model performs better.

Moving on to the nonlive recognizer evaluation, an internal word evaluation is conducted, which includes monophones, triphones, and tied-state triphones. The monophone, triphone, and triphone-based recognizers achieve accuracy rates of 78.12%, 86.87%, and 88.99%, respectively. Therefore, the triphone-based recognizer exhibits the best performance in nonlive recognition, as the comparison of internal word triphones enhances the recognizer's performance. Based on the comparison of results in both live and nonlive settings, it is recommended to focus on investigating Afan Oromo speech-based command and control in live scenarios.

Since there is no established speech corpus specifically for Afan Oromo computer MS commands, comparing the results of this research with previous studies is not meaningful. However, the existing literature on Amharic language indicates the investigation of speech computer commands with a maximum recognizer accuracy of 87% and 96% using HMM models with fixed variance, which is lower than the performance achieved in the Afan-Oromo speech-based computer command for nonparticipating users [39].

In conclusion, this research has demonstrated the feasibility of speech-based computer commands with promising performance using the available HMM tool under limited resources for Afan Oromo speech corpus.

5. Conclusions

Automatic speech recognition (ASR) involves converting speech into text to enable computers to understand and respond to human speech. Speech command-based interfaces have been implemented in various systems, including e-commerce, medical equipment, and digital devices, allowing users to control applications through speech input. However, there is currently no developed speech-based computer command interface for Afan Oromo.

The objective of this study was to investigate and develop an Afan Oromo speech-based command and control system using selected MS Word commands. The development process involved creating a speaker-independent, HMM-based Afan Oromo speech recognizer using the HTK toolkit. To collect data, speech recordings were obtained from 38 speakers (17 females and 21 males) between the ages of 18 and 40. Out of a total of 64 MS command words, 54 words were used for training (84.37%) and 10 words were for testing (15.63%). The performance of the recognizers was evaluated using both live and nonlive techniques.

In the nonlive recognizer evaluation, internal word evaluation was conducted, including monophones, triphones, and tied-state triphones. The monophone tied state, triphone, and triphone recognizers achieved accuracy rates of 78.12%, 86.87%, and 88.99%, respectively. Therefore, the triphone-based recognizer performed best in nonlive recognition. In the live setting, the recognizer's performance was assessed using fixed variance and variable variance models. The fixed variance model achieved a maximum accuracy of 91.82%, while the variable variance model performed at 96.29% for users who did not participate in the training.

Based on these results, it can be concluded that the variable variance model exhibits higher accuracy. In addition, the live recognizer outperformed the nonlive recognizer. However, the performance of the recognizer in a practical MS Office Word environment could not be evaluated due to the requirement of an object-as-a-service component for integration. The development of the speech interface prototype faced limitations in terms of language resources and tools, which affected the accuracy of the recognizer. Despite these challenges, the experiment yielded promising results, demonstrating the potential for developing a prototype for an Afan Oromo speech-based command and control system using selected MS Office words with fixed and variable models.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] H. H. O. Nasereddin and A. A. R. Omari, "Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation," in *Proceedings of the 2017 Computing Conference*, pp. 200–207, London, UK, July 2017.

[2] X. Sun, Q. Yang, S. Liu, and X. Yuan, "Improving low-resource speech recognition based on improved NN-hmm structures," *IEEE Access*, vol. 8, pp. 73005–73014, 2020.

[3] S. Saminu, G. Xu, Z. Shuai et al., "A recent investigation on detection and classification of epileptic seizure techniques using eeg signal," *Brain Sciences*, vol. 11, no. 5, p. 668, 2021.

[4] H. Satori, H. Hiyassat, M. Harti, and N. Chenfour, "Investigation Arabic speech recognition using CMU Sphinx system," *The International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 186–190, 2009.

[5] H. Ibrahim and A. Varol, "A study on automatic speech recognition systems," in *Proceedings of the 8th 2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, Beirut, Lebanon, June 2020.

[6] J. Howard, "Artificial intelligence: implications for the future of work," *American Journal of Industrial Medicine*, vol. 62, no. 11, pp. 917–926, 2019.

[7] N. T. Anh, Y. Hu, Q. He, T. T. N. Linh, H. T. K. Dung, and C. Guang, "LIS-Net: an end-to-end light interior search network for speech command recognition," *Computer Speech & Language*, vol. 65, Article ID 101131, 2021.

[8] H. Isyanto, A. S. Arifin, and M. Suryanegara, "Design and implementation of IoT-based smart home voice commands for disabled people using Google assistant," in *Proceeding of the 2020 International Conference on Smart Technology and Applications (ICoSTA)*, Surabaya, Indonesia, February 2020.

[9] C. Murad, C. Munteanu, B. R. Cowan, and L. Clark, "Revolution or evolution? Speech interaction and HCI design guidelines," *IEEE Pervasive Comput*, vol. 18, no. 2, pp. 33–45, 2019.

[10] R. S. Sharma, S. H. Paladugu, K. J. Priya, and D. Gupta, "Speech recognition in Kannada using HTK and julius: a comparative study," in *Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 68–72, Chennai, India, April 2019.

[11] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech," *Interspeech*, vol. 2019, pp. 2638–2642, 2019.

[12] K. R. Chowdhary, *Fundamentals of Artificial Intelligence*, Springer, Berlin, Heidelberg, 2020.

[13] R. Jahangir, Y. W. Teh, H. F. Nweke, G. Mujtaba, M. A. Al-Garadi, and I. Ali, "Speaker identification through artificial intelligence techniques: a comprehensive review and research challenges," *Expert Systems with Applications*, vol. 171, Article ID 114591, 2021.

[14] I. Tubert-Brohman, W. Sherman, M. Repasky, and T. Beuming, "Improved docking of polypeptides with glide," *Journal of Chemical Information and Modeling*, vol. 53, no. 7, pp. 1689–1699, 2013.

[15] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 6074–6078, 2020.

[16] B. V. Dyck, B. BabaAli, and D. V. Comperolle, "A hybrid ASR system for southern Dutch," *Computational Linguistics Jobs Netherlands*, vol. 11, pp. 27–34, 2021.

[17] H. Veisi and A. Haji Mani, "Persian speech recognition using deep learning," *International Journal of Speech Technology*, vol. 23, no. 4, pp. 893–905, 2020.

[18] T. K. Mohd, N. Nguyen, and A. Y. Javaid, "Multi-Modal Data Fusion in Enhancing Human-Machine Interaction for Robotic Applications: A Survey," 2022, http://arxiv.org/abs/2202.07732.

[19] A. G. D. Varda and C. Strapparava, "A layered bridge from sound to meaning: investigating cross-linguistic phonosemantic correspondences," *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 43, pp. 1029–1035, 2021.

[20] A. Sharma, A. Sharma, P. Juneja, and V. Jain, "Spectral features based speech recognition for speech interfacing to control PC Windows," in *Proceedings of the 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, pp. 341–345, Dehradun, India, August 2020.

[21] J. Schubert, J. Brynielsson, M. Nilsson, and P. Svenmarck, "Artificial intelligence for decision support in command and control systems," 2018, https://www.researchgate.net/publication/330638139_Artificial_Intelligence_for_Decision_Support_in_Command_and_Control_Systems.

[22] J. L. Dioses, "AndroiDuino-fan: a speech recognition fan-speed control system utilizing Filipino voice commands," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3042–3047, 2020.

[23] Y. T. Martha, *Application of Amharic Speech Recognition System to Command and Control Computer An Experiment with Microsoft Word*, Addis Ababa University, Addis Ababa, Ethiopia, 2003.

[24] B. Addis and S. Teferra, *Application of Amharic Speech Recognition System for Dictation in Judicial Domain*, Addis Ababa University, Addis Ababa, Ethiopia, 2021.

[25] A. Deksiso, *Spontaneous Speech Recognition for Amharic Using HMM*, Addis Ababa University, Addis Ababa, Ethiopia, 2015.

[26] R. M. Tamiru and S. T. Abate, "Sentence-Level Automatic Speech Segmentation for Amharic," *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer, Singaporepp. 477–485, 2021.

[27] T. A. Altaye, *Designing Automatic Speech Recognition for Ge'ez Language*, Addis Ababa University, Addis Ababa, Ethiopia, 2020.

[28] Universiti Malaysia Pahang Institutional Repository, "Isolated Malay Speech Recognition Using Fuzzy Logic," 2019, http://fypro.ump.edu.my/ethesis/index.php.

[29] Y. G. Gutu, *Continuous Speech Recognition System for Afaan Oromo*, Jimma University, Jimma, Ethiopia, 2016.

[30] D. D. Geleto, *Large Vocabulary Continuous Speech Recognition System for Afaan Oromo Using Hidden Markov Model (HMM)*, Adama Science Technology University, Adama, Ethiopia, 2016.

[31] P. Sudhakaran, A. K. Yadav, and S. Karamchandani, "Parasitic sorority of speech processing algorithms with an assortment of statistical toolkits," *J. Phys. Conf. Ser.*, vol. 1998, Article ID 012024, 2021.

[32] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 9411–9457, 2021.

[33] F. S. Al-Anzi and D. A. Zeina, "Performance evaluation of sphinx and HTK speech recognizers for spoken Arabic language," *Int. J. Innov. Comput. Inf. Control*, vol. 15, no. 3, pp. 1009–1021, 2019.

[34] M. Gales and S. Young, "The application of hidden Markov Models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2007.

[35] A. Akila and E. Chandra, "Isolated Tamil word speech recognition system using HTK," *Int. J. Comput. Sci. Res. Appl.*, vol. 3, no. 2, pp. 30–38, 2013.

[36] A. Choudhary, R. Chauhan, and G. Gupta, "Automatic speech recognition system for isolated; connected words of Hindi language by using hidden Markov model toolkit (HTK)," *International Conference on Emerging Trends in Engineering & Technology*, vol. 15, pp. 847–853, 2013.

[37] H. A. Elharati, M. Alshaari, and V. Z. Këpuska, "Arabic speech recognition system based on MFCC and HMMs," *Journal of Computer and Communications*, vol. 08, no. 03, pp. 28–34, 2020.

[38] A. Deksiso, *Spontaneous Speech Recognition for Amharic Using HMM*, Addis Ababa University, Adama, Ethiopia, 2015.

[39] K. Teshite, G. Mamo, and K. Calpotura, *Afan Oromo Speech-Based Computer Command and Control: An Evaluation with Selected Commands*, Jimma University, Jimma, Ethiopia, 2023.