ANALYSIS OF AUDIO DATA TO MEASURE SOCIAL INTERACTION IN THE TREATMENT OF AUTISM SPECTRUM DISORDER USING SPEAKER DIARIZATION AND IDENTIFICATION

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

AKSHAY CHAVAKULA

Dr. Fang Wang, Thesis Supervisor

May 2021

The undersigned, appointed by the Dean of the Graduate School, have examined

the thesis titled:

ANALYSIS OF AUDIO DATA TO MEASURE SOCIAL INTERACTION IN THE TREATMENT OF AUTISM SPECTRUM DISORDER USING SPEAKER DIARIZATION AND IDENTIFICATION

presented by Akshay Chavakula,

a candidate for the degree of Master of Science, and hereby certify that,

in their opinion, it is worthy of acceptance

_____

Dr. Fang Wang

_____

Dr. Dong Xu

_____

Dr. David Beversdorf

ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance of my supervisor, and I would like to express my profound gratitude to Dr. Fang Wang for her support, and encouragement. She pushed me to explore new areas, challenged my thinking and ultimately helped me grow as a student and a researcher. I am fortunate to have Dr. Fang as my faculty advisor.

I would like to thank Dr. David Beversdorf for collaborating with us on this research study and serving as a member on my thesis committee.

I would also like to thank Dr. Dong Xu for his service as a member on my thesis committee.

Furthermore, I would like to thank our research team, Dr. Bradley Ferguson, Youngbin Ha, Amy Costa and Thunpisit Amnuaikiatloet for working with me and helping me throughout this research study.

I would like to express my sincere thanks to the IT program for providing me with research and teaching assistantship(s) during my graduate studies here at the University of Missouri-Columbia.

Finally, I would like to acknowledge that I will always be indebted to my family for having faith in me and pushing me to excel while exploring my interests.

Columbia, Missouri

April 13, 2021

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure **Page**

ABSTRACT

Autism spectrum disorder (ASD) is a neurodevelopmental disorder that affects communication and behavior in social environments. Some common characteristics of a person with ASD include difficulty with communication or interaction with others, restricted interests paired with repetitive behaviors and other symptoms that may affect the person's overall social life. People with ASD endure a lower quality of life due to their inability to navigate their daily social interactions. Autism is referred to as a spectrum disorder due to the variation in type and severity of symptoms. Therefore, measurement of the social interaction of a person with ASD in a clinical setting is inaccurate because the tests are subjective, time consuming, and not naturalistic.

The goal of this study is to lay the foundation to passively collect continuous audio data of people with ASD through a voice recorder application that runs in the background of their mobile device and propose a methodology to understand and analyze the collected audio data while maintaining minimal human intervention. Speaker Diarization and Speaker Identification are two methods that are explored to answer essential questions when processing unlabeled audio data such as who spoke when and to whom does a certain speaker label belong to?

Speaker Diarization is the process of partitioning an audio signal that involves multiple people into homogenous segments associated with each person. It provides an answer to the question of "who spoke when?". The implemented Speaker Diarization algorithm utilizes the state-of-the-art d-vector embeddings that take advantage of neural networks by using large datasets for training so variation in speech, accent, and acoustic conditions of the audio signal

can be better accounted for. Furthermore, the algorithm uses a non-parametric, connection-based clustering algorithm commonly known as spectral clustering. The spectral clustering algorithm is applied to these previously extracted d-vector embeddings to determine the number of unique speakers and assign each portion of the audio file to a specific cluster.

Through various experiments and trials, we chose Microsoft Azure Cognitive Services due to the robust algorithms and models that are available to identify speakers in unlabeled audio data. The Speaker Identification API from Microsoft Azure Cognitive Services provides a state-of-the-art service to identify human voices through RESTful API calls. A simple web interface was implemented to send audio data to the Speaker Identification API which returned data in JSON format. This returned data provides an answer to the question – "who does a certain speaker label belong to?".

The proposed methods were tested extensively on numerous audio files which contain various numbers of speakers who emulate a realistic conversational exchange. The results support our goal of digitally measuring social interaction of people with ASD through the analysis of audio data while maintaining minimal human intervention. We were able to identify our target speaker and differentiate them from others given an audio signal which could ultimately unlock valuable insights such as creating a bio marker to measure response to treatment.

CHAPTER

1. Introduction

Most life forms have built in, biologically adapted forms of communication. The human voice is unique, and our planet consists of 7.5 billion distinct human voices. Speaker Diarization and Speaker Identification are two methods that are explored to differentiate the target speaker's voice from the others in an audio signal. This facilitates the answers to questions such as who spoke when and to whom does a certain speaker label belong to? Smart devices are the portal through which data is mined to gather an abundant amount of information. The collected data can be further analyzed to create a bio marker to measure response to treatment.

1.1 Goal of Study

The goal of this study is to passively collect continuous audio data from the daily social interactions of people with autism spectrum disorder (ASD) to accurately differentiate and identify the target speaker given an unlabeled audio signal. The collection of continuous data from people with ASD is done through their own mobile or wearable device which can be analyzed to differentiate and identify the target speaker from others in an unlabeled audio signal.

- Passively collect continuous audio data through a minimally invasive application that runs in the background and actively stores the audio files onto a cloud storage service known as Firebase storage.

- Differentiate the target speaker from others to accurately determine who spoke when with precise time stamps for the duration of the audio data.
- Identify the target speaker in an unlabeled audio signal to determine when a person with ASD spoke.
- Develop a methodology to reduce manual human intervention in the measurement of social interaction of people with ASD.
- Automate the process of collecting and analyzing social interaction outside of a clinical setting.
- Lay the groundwork for future improvement and expansion of the Speaker Diarization and Speaker Identification methods.

2. Literature Review

A review of the current work done in the field is discussed in this section. Three research papers from the industry and other research institutions relative to this study are examined.

2.1 A Convolutional Neural Network Smartphone App for Real-Time Voice Activity Detection

A smartphone application developed for real-time voice activity detection is discussed in this paper. Voice activity detectors are used to differentiate speech from the non-speech portions in an audio signal. Likewise, in this instance, a VAD is used to differentiate the speech from the noise when given a noisy speech signal. The VAD pipeline begins with taking an audio signal with noise as an input then the estimation and the classification of the noise is performed to reduce the noise in the signal and ultimately output the speech signal without the noise. This is very beneficial in hearing aids due to the promising increase in performance. One key observation to note here is that this VAD must be implemented in real time and in a frame-based manner which is developed to run on smartphones. This paper further addresses challenges that arise with a real time implementation such as computational efficiency, frame processing rate and accuracy in realistic scenarios using deep learning approaches such as a convolutional neural network. A common issue with real-time implementation of convolutional neural networks such as slow inference time is addressed in this work. A sampling rate of 48 kHz with a buffer size of 1.34 milliseconds is used to reduce the latency associated with frame-based audio processing. A

multi-threaded approach is used for the CNN implementation where the CNN is run on a parallel synchronous thread while the image formation is being executed on the main audio i/o thread because input images to the convolutional neural network must be extracted on frame-by-frame basis, but the classification is not required to be done per frame basis. As a result, this allows more efficient usage of the computation time in main audio i/o thread for the execution of other modules in the speech processing pipeline. The proposed smartphone application in this paper is compared to other voice activity detector applications and algorithms. The author claims that the result from the experimentation shows that the proposed smartphone application using a deep learning approach has very good performance. This paper discusses the advantages and reasoning behind using a smartphone application by exploiting the existence of powerful multi core processors and low latency Bluetooth connectivity to other hardware such as hearing devices to the smart phones owned by more than three quarters of the United States population.

2.2 Android Voice Recognition Application with Multi Speaker Feature

Speech processing using Mel Frequency Cepstral Coefficients and the basis of voice characteristics along with an Android application that can perform both single speaker and multi speaker identification is discussed in this paper. Mel-Frequency Cepstrum, referred to as MFC, is used in the field of sound processing as a representation of the short-term power spectrum of a sound. MFC is obtained from a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. A Mel Frequency Cepstral Coefficient (MFCC) is essentially a coefficient that collectively makes up an MFC. MFCCs are derived in the following method: first

a Fourier transform of an audio signal is taken. Next, the power of the spectrum obtained on a mel scale using triangular overlapping windows is mapped. After that, the log of the powers at each of the mel frequencies is taken. Next, the discrete cosine transform of the list of mel log powers is taken as if it were a signal. Finally, the amplitudes of the resulting spectrum are the MFCCs. This work handles the problem of voice recognition on the software end and claims to achieve a high level of accuracy. Multi speaker identification feature can identify more than one speaker by segmenting the input into smaller segments and treating each segment as a single speaker identification problem. The segmentation is handled in the time domain where given a sample of N seconds, the sample is segmented into K different segments where the smallest segment is greater than 1 second. The segmented portion is passed to the speaker identification algorithm. The algorithm iterates over the K number of segments then stores the identified speakers with a similarity score greater than or equal to the minimum acceptable threshold. Next, the algorithm iterates over the entire sample again but this time it reduces the number of segments by 1. The algorithm keeps reducing the number of segments until it iterates over the entire N second sample in the final iteration. The proposed algorithm in this paper claims to correctly identify individual speakers 90.3% of the time and can correctly identify up to 3 individuals 86% of the time. The average identification time was around 220 milliseconds for 415 different voice samples with users from both male and female genders.

2.3 Speaker Recognition Using Mel Frequency Cepstral Coefficient and Locality Sensitive Hashing

In this paper, various traditional speaker recognition models are compared to this newly proposed model using feature extraction through MFCCs and applying a big data classification method to these extracted features known as Locality Sensitive Hashing (LSH). Speaker recognition pertains to the process of identifying individuals based on their voice characteristics. A speaker recognition model can be split into two main sections which contain feature extraction and training based on the extracted features. Feature extraction generally produces Mel Frequency Cepstral Coefficients (MFCCs) or Linear Predictive Cepstral Coefficients known as LPCCs. The proposed method in this papers extracts features from a speech signal using Mel Frequency Cepstral Coefficients (MFCCs) feature vectors which are then used to generate an acoustic speech signal. A common classifier in big data problems is known as Locality Sensitive Hashing referred to as LSH. The authors of this paper claim that this new model is more robust, effective, and accurate compared to traditional models that pair MFCC with Gaussian Mixture Model (GMM), another feature extraction method known as Linear Predictive Cepstral Coefficient paired with GMM, and MFCC paired with Probabilistic Neural Network (PNN). First, the MFCC features are extracted from an audio signal then the LSH classifier is applied to these extracted features to generate a hash table. One key takeaway from this paper is that they used dual categories of classifier like GMM and LSH and they found that 16 mixtures is the ideal number of Gaussians when using a Gaussian Mixture Model (GMM). The Locality Sensitive Hashing (LSH) method is constructed according to theory where it uses mini hashing to compress large scale data into small signatures.

3. Background

Background information about autism spectrum disorder and acoustics which are leveraged by a speaker recognition system are discussed in this section.

3.1 Autism Spectrum Disorder

Autism spectrum disorder, referred to as ASD, is a neurodevelopmental disorder. People with ASD generally show deficiency in social skills, communication, repetitive behaviors, and interests. The greatest challenge that most people with ASD face is navigating their daily social interactions which reduces their overall quality of life. There have been a lot of studies and efforts in the past that focus on social interactions and interventions in a clinical setting. However, a person with ASD visits a doctor occasionally so any data that is collected is sparse and limited to their visit. Furthermore, the data is recorded and processed manually which is labor intensive and unorganized. Therefore, measuring social interaction in a clinical setting is time consuming and inaccurate. The collection of continuous data during real social interactions of a person with ASD is greatly beneficial in the creation of a bio marker to aid the outcome measurement for treatment. However, the problem of measuring social interaction outside of a clinical setting remains a challenge. The automation of social interaction measurement outside of a clinical setting reduces the required manual human intervention which helps the data quality through removing bias and human error.

3.2 Acoustics

Speaker recognition utilizes and evaluates the vital acoustic features of speech which are unique to everyone. These acoustics features differ from person to person based on their physical characteristics such as the shape and size of a person's mouth and throat. Each person has a distinct speech style and unique voice pitch which are leveraged by speaker recognition to recognize and classify individuals based on their voice characteristics. Speaker recognition has been classified as a behavior biometric due to the scientific progress made in this field in the recent years. The evaluation of these vital acoustic features is at the core of a speaker recognition system.

4. Methodology

4.1 Speaker Diarization

Speaker diarization is the process of partitioning an audio signal that involves multiple people into homogenous segments associated with each person. Moreover, it is the process of recognizing “who spoke when”. Speaker diarization has been a well-known problem for many years and it plays a crucial role in every acoustic speech recognition system (ASR). Speaker diarization was considered an upstream processing step for a speech recognition system in the past. However, in the recent years, speaker diarization has become a standalone problem to solve.

Some common uses cases for speaker diarization include analyzing medical conversations, video captioning, and call center or meeting transcriptions. Diarization maps a segment of speech such as a word or a sentence into a space that represents a speaker’s voice characteristics then clusters the segment representations. A key problem to investigate is how to efficiently map the speech segments to the representation space so that different speakers are accurately mapped to different positions in the space.

A complete speaker diarization system consists of the following modules - speech detection, speech segmentation, embedding extraction, clustering, and transcription (optional). A speech detection and segmentation module are used to differentiate speech from the non-speech portions in an audio signal. In other words, this will remove all the silences and noise in the audio file then it will divide the input utterance into small segments. An embedding extraction module is

used to extract speaker discriminative embeddings such as i-vectors or d-vectors from the small segments. A clustering module determines the number of speakers and the assignment of speaker identities to each segment. Finally, a re-segmentation module further refines the diarization results by enforcing additional constraints to produce the final diarization output.

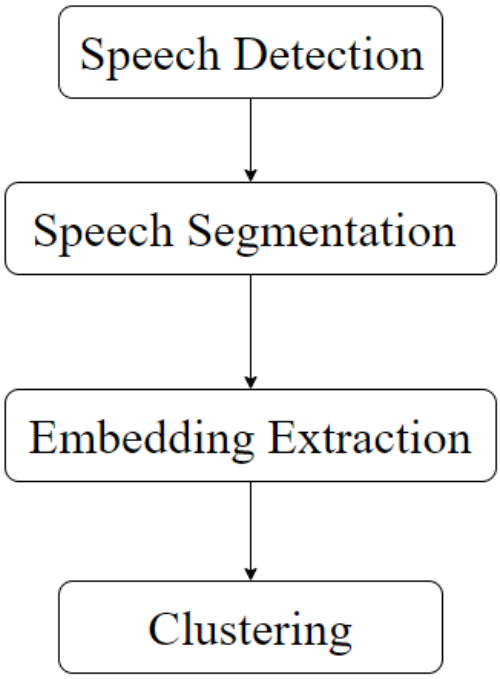

Figure 4.1.1 Speaker Diarization Modules Flow Chart

The identity vector known as an i-vector is the spectral signature for a particular slice of speech. The extraction of an i-vector is essentially a dimensionality reduction of the Gaussian Mixture Model (GMM) supervector. Therefore, this framework assumes that the i-vector has a gaussian distribution. A d-vector is extracted using a deep neural network (DNN) which is trained to take

stacked filterbank features and generate a speaker probability on the output. A d-vector is the averaged activation from the last hidden layer of this DNN. Unlike the i-vector framework, d-vector framework does not have any assumptions about the features' distributions. The main advantage of using d-vectors is that neural networks can be trained using large datasets that can account for variation in speech, accent, and acoustic conditions of the speaker's environment. Recent studies have discovered that diarization performance improves significantly by replacing the i-vectors with the neural network embeddings known as d-vectors.

Clustering is an unsupervised machine learning method that creates clusters or groups of the data in a n-dimensional space. Spectral clustering is a popular clustering method used for the speaker diarization problem. Spectral clustering is treated as a graph partition problem because data points are treated as nodes of a graph and these nodes are then mapped to a low dimensional space that can be easily segregated to form clusters. Spectral clustering creates an affinity matrix of the data. An affinity matrix, also known as similarity matrix, organizes mutual similarities between a set of data points.

Recent work has shown that using d-vector embeddings has significantly improved the performance of an enrollment-verification two-stage application. Translating this work to speaker diarization promises improved performance compared to the previously used method with i-vector embeddings. Given an audio file, the first step is to transform the audio signals into frames with a set width and step parameters. Then log-mel-filterbank energies of dimension 40 are extracted from each frame and used as an input to the LSTM-based network. Log-mel filterbank energies are used to capture the variations in speech such as pitch, quality, and tone of voice. They are obtained by performing a discrete Fourier transform (DFT) on the speech signal. LSTM stands for long short-term memory and it is an artificial recurrent neural network

architecture used in the field of deep learning. LSTM is unique compared to other traditional neural network architectures because it uses feedback connections unlike the standard feedforward neural networks. One main advantage is that it can process an entire sequence of data rather than a single data point. Next, fixed length sliding windows are constructed on these frames and the LSTM-based network is executed on each window. The last frame output of the LSTM-based network is used as a d-vector representation of this sliding window.

A voice activity detector (VAD) is necessary to differentiate speech from the non-speech portions of an audio signal. These speech segments are further divided into smaller non-overlapping segments using a maximal-segment length limit, usually around 400 milliseconds, which determines the temporal resolution of the diarization results. The temporal resolution refers to the discrete resolution of a measurement with respect to time. It is defined as the amount of time needed to revisit and acquire data from the exact same location. Then, the corresponding d-vectors are L2 normalized and averaged to form an embedding for each segment. Next, a clustering algorithm is applied to these embeddings to determine the number of unique speakers and assign a specific speaker label to each portion of the entire audio file.

There are many clustering algorithms that can be utilized for speaker diarization. Some popular clustering methods include links clustering, k-means clustering, and spectral clustering. One key difference to note is online versus offline clustering. In online clustering, a speaker label is immediately produced once a segment is available without processing the future segment. Whereas, in offline clustering, speaker segments are produced after the embeddings of all the segments are processed. Therefore, offline clustering algorithms tend to outperform online clustering algorithms due to the contextual information available in the offline setting. Naïve online clustering is a technique where each cluster is represented by the centroid of all its

corresponding embeddings. Each new segment embedding is compared to the centroids of all other existing clusters to determine the similarity in this method. A new cluster containing only the new segment embedding is created when it's smaller than a minimum threshold. Otherwise, it will be added to the cluster with the greatest similarity. Google LLC and Carnegie Mellon University collaborated on a research paper titled "Speaker Diarization with LSTM" [4] to develop a new clustering method named Links online clustering. Links online clustering is built on the naïve online clustering method by estimating the cluster probability distributions and the substructure is modeled based on the embedding vectors. However, the more popular clustering methods for speaker diarization are offline clustering algorithms such as k-means and spectral offline clustering.

There are three main properties of speech data that causes k-means clustering to perform poorly – non-gaussian distributions, cluster imbalance, and hierarchical structure. Speech data in most situations is non-gaussian so therefore it is difficult to use the centroid of a cluster for accurate representation. Cluster imbalance is introduced due to the difference in speech time between different speakers. K-means clustering may incorrectly segment large clusters into smaller clusters because one speaker might dominate most of the audio file while other speakers speak less frequently. There are various speaker characteristics that differentiate speakers such as age, gender, tone, and pitch. However, some characteristics are more difficult to differentiate than others. For example, differentiating between a male and a female speaker is easier than differentiating between two female speakers. Due to the nature of these differences, k-means clustering often struggles in performance. For instance, the k-means clustering algorithm often clusters all the male speakers into one cluster and all the female speakers into another cluster.

These problems can be mitigated by using a non-parametric connection-based clustering algorithm like spectral clustering.

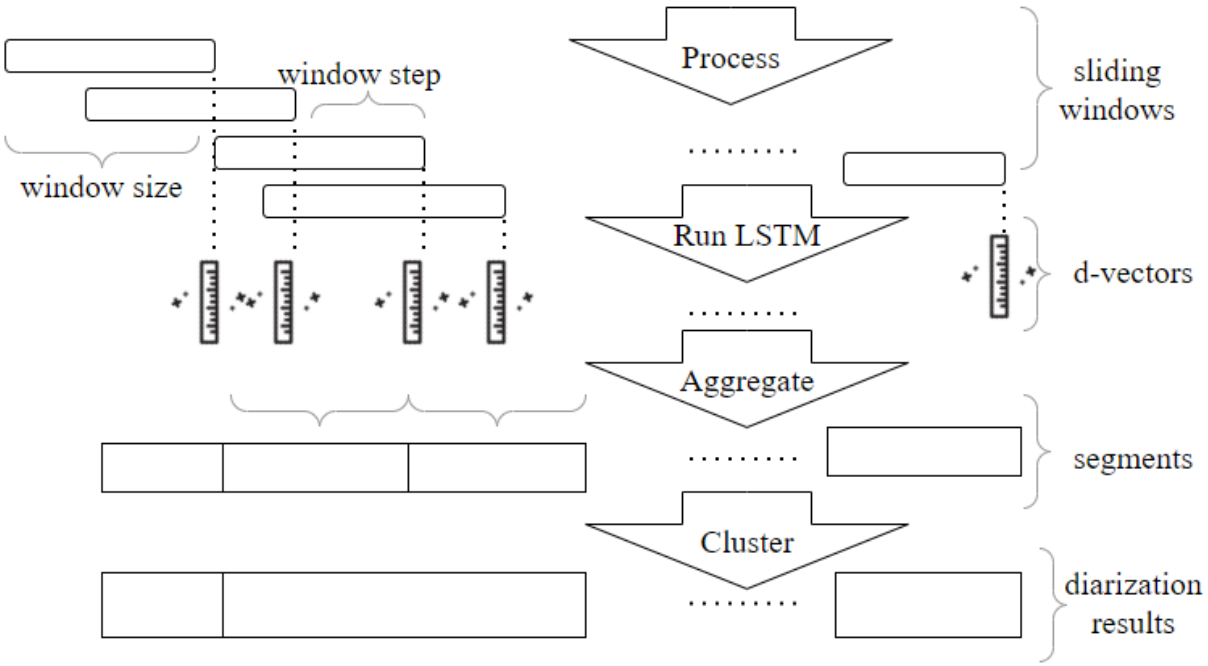

Figure 4.1.2 d-vector based Speaker Diarization System [4]

4.2 Microsoft Azure Cognitive Services

The Speaker Recognition API in Microsoft Azure Cognitive Services provides state-of-art algorithms which are easy to implement. These cognitive cloud APIs have many real-world use cases such as biometric voice authorization within applications containing sensitive data, real

time speech recognition, and they help identify individuals based on their unique voice characteristics.

The Speaker Recognition API in Microsoft Azure Cognitive Services uses machine learning and artificial intelligence to provide robust services that identify individual speakers and allow users to use speech for authentication purposes. Speaker recognition can be divided into two categories – verification and identification. Speaker verification is the process of authenticating users based on their unique voice characteristics. An interesting use case for this would be biometric authentication for applications carrying sensitive information such as financial or health care data. This has also been implemented in the latest smart phones where the voice assistant only responds to the commands from the owner of the device. Speaker identification is the process of determining the identity of a speaker by comparing the voice characteristics from the speech signal in the recognition audio file to the voice characteristics of each speaker in a list of the prospective pre-enrolled speakers. The Speaker Recognition API provides algorithms that verify and identify human voices using secure, RESTful API calls. A key observation to note is the difference between speaker recognition and speech recognition. Speech recognition pertains to determining the content of the audio file whereas speaker recognition is the process of identifying the speaker of the content.

4.2.1 Speaker Verification

Speaker verification is the process of verifying an enrolled speaker’s identity with either pre-set passphrases or free form voice input. In other words, speaker verification checks the likelihood

that two different speech signals belong to the same person. Verification API uses artificial intelligence to authenticate users and improve security for sensitive resources. Speaker verification can be either text-dependent where speakers choose a certain passphrase for both the enrollment and verification phases or it can be text-independent where speakers can speak freely as they do in a real-word setting. In text-dependent verification, the speaker's voice is enrolled by saying a set of predetermined passphrases from which voice characteristics are extracted to form a unique voice signature. This voice signature and the selected passphrase that the speaker used in the enrollment phase are used together to verify the identity of a certain speaker. In text-independent verification, the speaker can speak freely for a set period from which the voice characteristics are extracted to form a unique voice signature which is used to compare to the voice characteristics extracted from the recognition audio to verify the identity of the speaker.

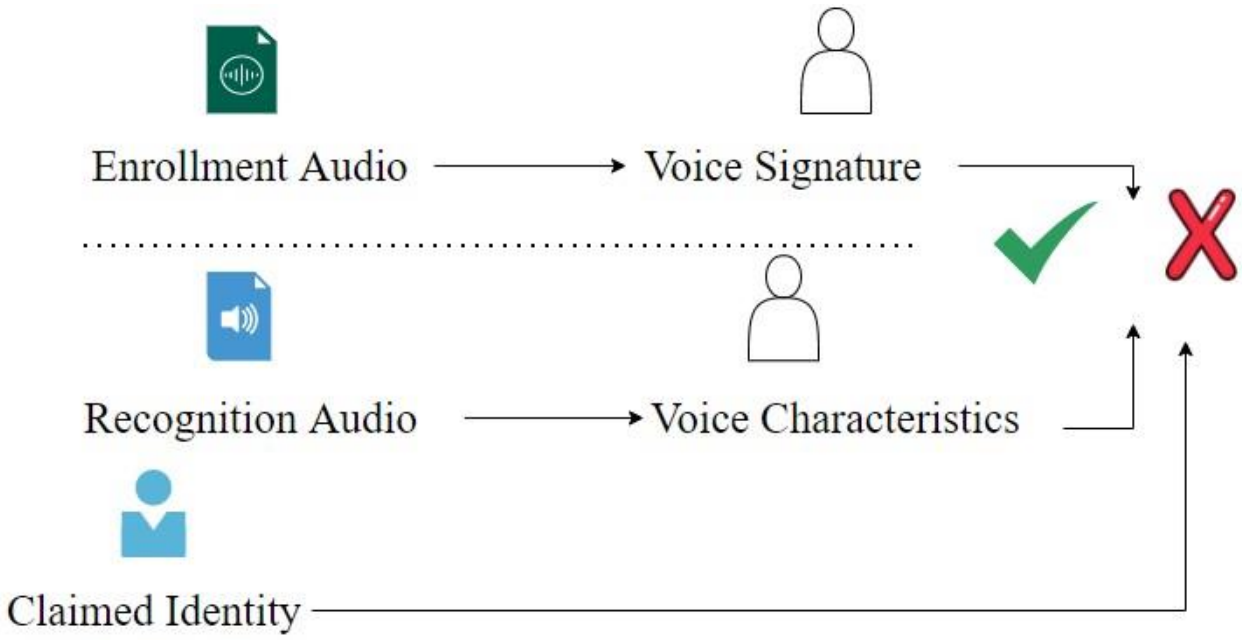

Figure 4.2.1 Speaker Verification Process Flow

4.2.2 Speaker Identification

Speaker Identification is the process of determining the identity of an unknown speaker by comparing the voice characteristics extracted from a recognition audio file to the voice characteristics of each speaker in a list of enrolled profiles. This enables the identification of individual speakers in a multi-speaker conversational setting. Enrollment for speaker identification is text-independent so the speaker can speak freely for a period from which their voice characteristics are extracted, assigned a unique id, and stored within a list of pre-enrolled speakers. The voice characteristics of each speaker from the list of enrolled profiles are compared to the voice characteristics extracted from the recognition audio file then each profile in the list is assigned a similarity score based on how similar the voice characteristics of the certain enrolled profile is to the voice characteristics from the recognition audio file. There can be up to 50 enrolled speakers per each request. The speaker identification API supports various languages such as English, French, Spanish, Chinese, German, Italian, and Portuguese. The accept and reject outcomes vary based on the situation and type of data being analyzed so Microsoft Azure allows the developers to customize the accept/reject threshold, which by default is set to 0.50, to determine what level of similarity is acceptable to accurately identify an individual. This means enrolled profiles that achieve a similarity of score of greater than 0.50 will be identified as the speakers in the recognition audio file.

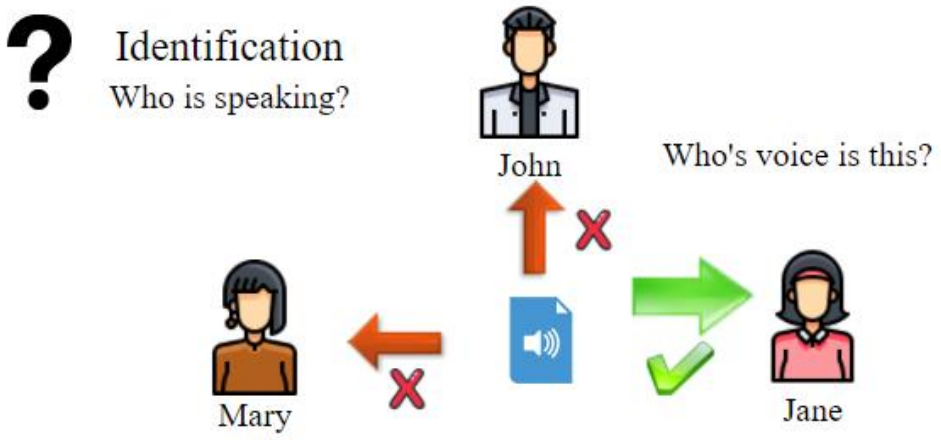

Figure 4.2.2.1 Speaker Identification

Identification requires a recognition audio file and a list of profiles containing potential speakers for comparison.

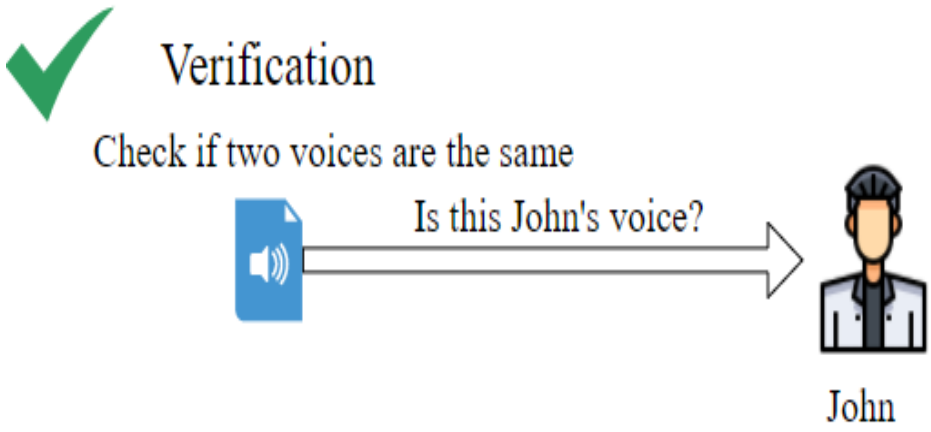

Figure 4.2.2.2 Speaker Verification

Verification only requires a recognition audio file and a speaker profile for comparison.

4.2.3 Enrollment

Both speaker verification and speaker identification require existing voice samples for comparison. Enrollment refers to the process of pre-recording a speaker's voice to extract the acoustic and speech patterns that form a voice print. Each person has a unique voice print like how each person has a unique fingerprint. A voice print is sometimes referred to as a voice template or a voice model.

Speaker Identification is more complicated since it requires more comparisons to voice prints than Speaker Verification. For example, Speaker Verification compares the voice characteristics from the recognition audio file to a single voice print to determine if the two voices belong to the same speaker. Whereas Speaker Identification tries to identify speakers from the recognition audio. This requires comparison of the voice characteristics from the recognition audio to the voice prints of each of the speaker in the list of enrolled profiles to determine the most similar profile.

The enrollment process for verification and identification are separate because the algorithms behind verification and identification are different. There are two types of speaker recognition systems – text-dependent and text-independent. Text-dependent is generally used for verification enrollment where prompts are known and standard across all speakers. Text-independent is used for identification enrollment where speakers can provide free form voice input in a specified language for a period. Since identification does not require the comparison of what was said

during the enrollment phase, text-independent enrollment can execute without the speaker's knowledge. Text-dependent enrollment is more controlled of these methods and it is at the core of Speaker Verification.

4.2.4 Speaker Recognition Process

The following three steps are required to start recognizing users with the Speaker Recognition API – profile creation, enrollment, and recognition. The first step is to create a profile to represent the user that we want the Speaker Recognition API to recognize. The next step is to enroll the user using text-dependent or text-independent enrollment. The optimal enrollment audio length for Speaker Identification is between 10 seconds and 5 minutes. This would be text-independent since the speaker can provide free from voice input for a given amount of time. Generally, Speaker Verification uses text-dependent enrollment since it requires the user to repeat a specific verification phrase three times. Recognition is possible once the enrollment process is completed. Identification requires an audio sample as well as a list of profiles containing potential speakers to compare to. While verification only requires an audio sample and a single speaker profile to compare and verify if the voice does or does not belong to a specified user.

4.2.5 Security and privacy

Although the details of the storage system are not accessible to the public, Microsoft ensures that the speaker enrollment data is stored in a secured system. This includes the speech audio for enrollment and the extracted voice signatures. After the initial enrollment, the enrollment audio is only used when the algorithm or model is updated, and the re-extraction of voice features is necessary. Microsoft does not store the audio data that is sent for recognition purposes. The developers have the freedom to create, update or delete data through secure, RESTful API calls. All the stored data is deleted when an Azure subscription is terminated.

4.2.6 Cognitive Services Pricing – Speaker Recognition

Users are only billed for their usage so there are no upfront costs or termination fees. Table 4.2.6 gives a detailed breakdown for the Speaker Recognition API pricing. We are currently using the free instance since we can manage with less than 10,000 transactions per month. One important takeaway here is that the Speaker Identification costs are nearly twice as much as Speaker Verification costs.

Instance	Transactions per Second (TPS)	Features	Price
Free	20 per minute	Speaker Verification Speaker Identification	10,000 transactions free per month
Standard	5 TPS	Speaker Verification	• 0-50K Transactions - $5 per 1,000 transactions • 50K-100K Transactions - $4.50 per 1,000 transactions • 100K-250K Transactions - $4 per 1,000 transactions • 250K-500K Transactions - $3.50 per 1,000 transactions • 500K+ Transactions - $3 per 1,000 transactions
		Speaker Identification	• 0-50K Transactions - $10 per 1,000 transactions • 50K-100K Transactions - $9 per 1,000 transactions • 100K-250K Transactions - $8 per 1,000 transactions • 250K+ Transactions - $7 per 1,000 transactions

Table 4.2.6 Microsoft Cognitive Services Pricing – Speaker Recognition

5. Implementation

This section covers the implementation of both the Speaker Diarization and Speaker Identification methods. These two methods serve different purposes, and the implementation of these methods is separate.

5.1 System Architecture

The process starts with data collection through mobile or wearable devices. The collected data is stored on a cloud storage service known as Firebase storage. The data from the database is then processed by both the Speaker Diarization and Speaker Identification methods. The results from both these methods will be different since they serve different purposes.

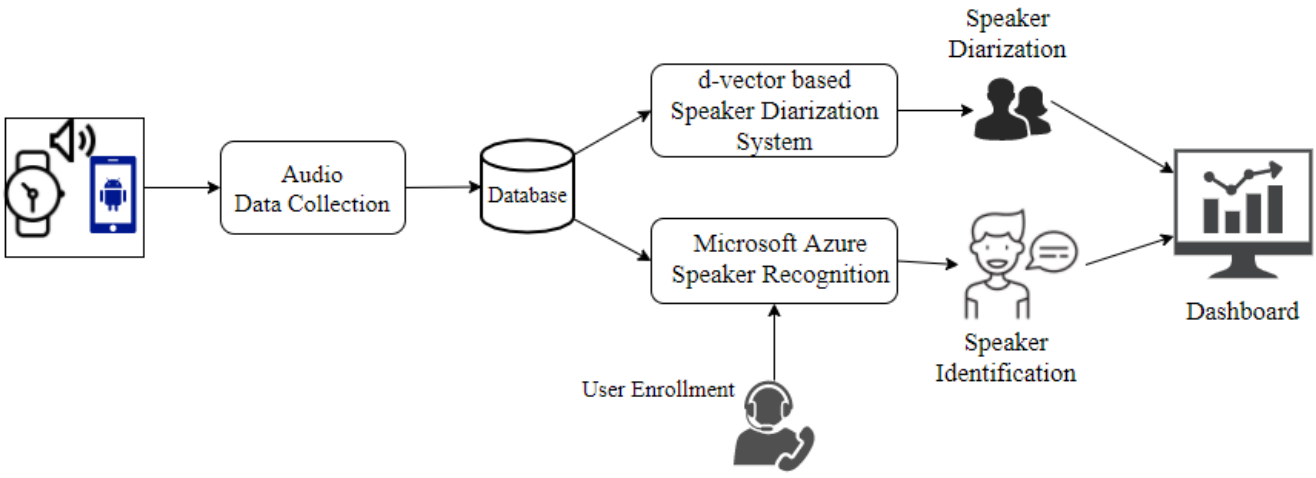

Figure 5.1 System Architecture

5.2 Speaker Diarization Overview

The speaker diarization algorithm can be segmented into four main modules – speech detection, speech segmentation, embedding extraction, and clustering. In the speech detection module, a voice activity detector (VAD), which is a neural network trained to differentiate speech from non-speech signals, is used to trim out all the silences/noise from an audio signal. Next, in the speech segmentation module, the audio file is segmented into windows with overlap. Specific segments of the audio file are magnified, and the size of the window determines the size of the segment. For example, if the window size is five seconds and the overlap is set to two seconds then the first window would start at zero seconds and stop at five seconds then the next window would start at two seconds and stop at seven seconds, so on and so forth until the entire audio file is properly segmented. After that, embeddings are extracted from each of these previously determined segments. The MFCC (Mel Frequency Cepstral Coefficient) of each of the segments is extracted by performing a discrete Fourier Transform (DFT) on the speech signal which accounts for the variations in speech including the pitch and tone of the voice. The SciPy library in python has a module that is utilized for extracting the MFCCs for each audio segment. Next a LSTM-based network, which uses these extracted MFCCs as input, outputs a vector representation which is properly known as a d-vector. The final step is to apply the spectral clustering algorithm these d-vector embeddings to create clusters of the data in a n-dimensional space, determine the number of unique speakers and assign each portion of the audio file to a specific speaker.

5.3 Speaker Diarization Algorithm

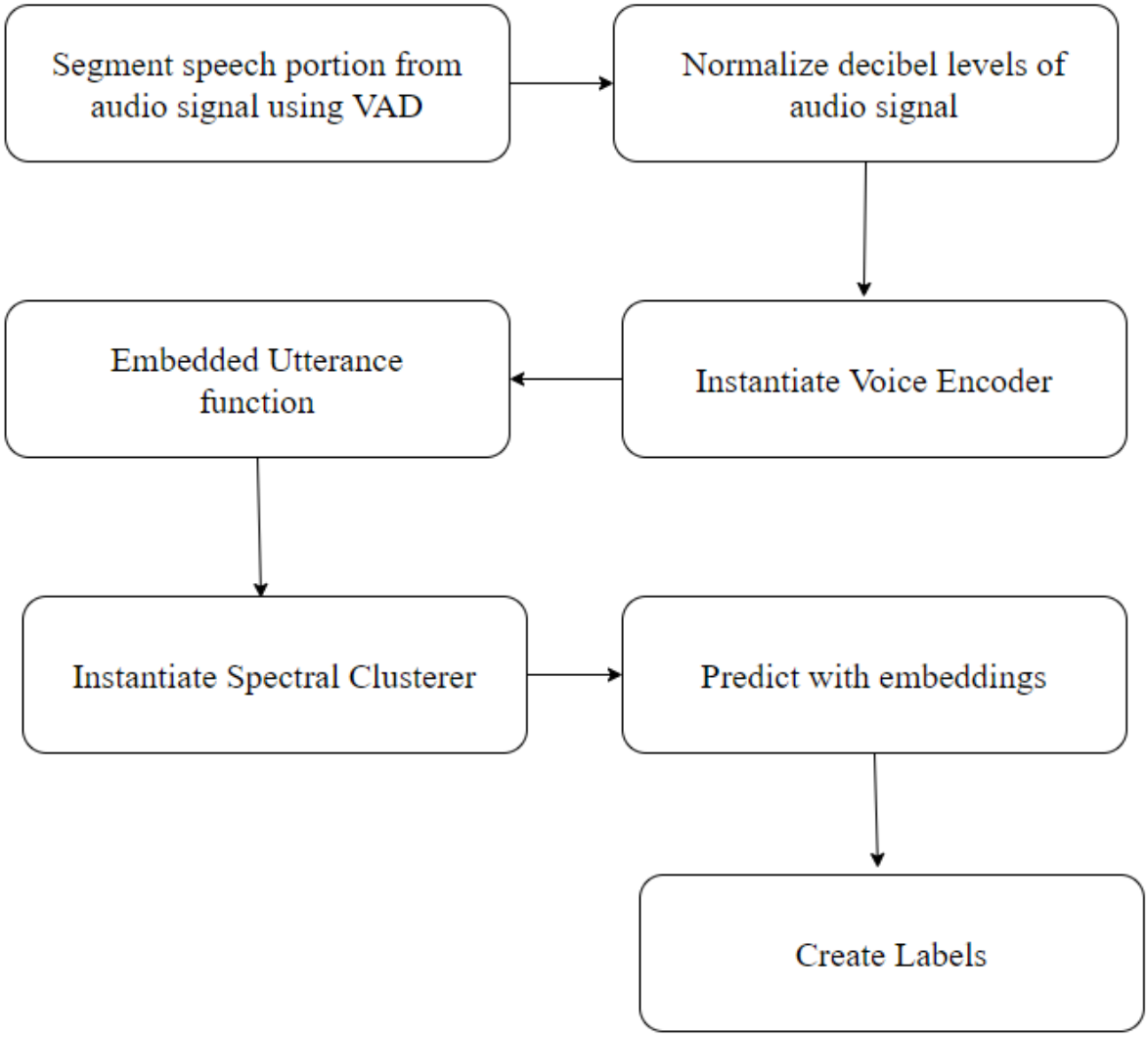

Figure 5.3.1 Speaker Diarization Algorithm Process Flow

The prerequisites for implementing the Speaker Diarization algorithm include getting Anaconda 1.7.2, Python 3.8.5, cloning an open-source repository known as Resemblyzer and importing Python libraries such as webrtcvad, librosa, PyTorch, torchvision and spectral clusterer.

Resemblyzer will be utilized in the speech detection, speech segmentation and embedding extraction modules of the Speaker Diarization algorithm. Resemblyzer computes a high-level representation of a speaker's voice using a deep learning model known as Voice Encoder. It takes an audio file as an input then creates a summary vector of 256 values that summarize the voice characteristics of the speaker. This summary vector is referred to as an embedding. Spectral clusterer is an open-source implementation of the spectral clustering algorithms provided by one of the authors from the "Speaker Diarization with LSTM" [4] paper. The first step in the speaker diarization algorithm is to preprocess the audio file using a voice activity detector (VAD) to trim out all the silences and the non-speech portions in an audio signal. The preprocessing step also includes normalizing the decibel level of the audio file. Next, an instance of the Voice Encoder class named encoder is created then "cpu" or "gpu" can be passed in as the parameter so the default device will be set for the algorithm to utilize based on the specifications of your system. The embedded utterance function of this instance takes the preprocessed audio file then segments it into windows, makes MFCCs (Mel Frequency Cepstral Coefficients), and creates d-vectors of these audio segments. A n by d matrix named continuous embeddings is created where n is the number of segments created which is equal to the number of d-vectors and d is the dimension of each d-vector which by default is 256. A list is also created that tracks the start and end times of each window for which a d-vector has been created. Now, the spectral clustering algorithm specifications can be configured relative to the audio file by setting the minimum, maximum number of clusters, p percentile, and gaussian blur. The previously created continuous embeddings matrix is passed in as a parameter to the predict function of the spectral clusterer to create labels. The final step is to create the labelling tuples, which is a list of finite ordered pairs, in (speaker label, start time, end time) format for the entire audio file. Now we can

determine which speaker spoke when for the entire duration of the audio file. The next challenge to solve is to find the answer to which speaker does a certain speaker belong to and this is where the speaker identification system is essential.

```
[('1', 0, 1.1), ('0', 1.1, 5.72), ('1', 5.72, 9.98), ('2', 9.98, 18.98), ('0', 18.98, 33.02), ('1', 33.02, 43.52), ('2', 43.52,
51.2), ('1', 51.2, 52.4), ('2', 52.4, 59.48), ('0', 59.48, 63.14), ('1', 63.14, 67.22), ('0', 67.22, 67.34), ('1', 67.34, 67.
4), ('0', 67.4, 71.42), ('1', 71.42, 76.04), ('2', 76.04, 79.82), ('0', 79.82, 84.86), ('2', 84.86, 89.6), ('0', 89.6, 95.84),
('1', 95.84, 101.54), ('2', 101.54, 107.9), ('1', 107.9, 109.76)]
```

Figure 5.3.2 Speaker Diarization Algorithm Output

5.4 Speaker Identification

A simple, easy-to-use web interface was built using HTML5, CSS3 and JavaScript to send audio data to the Speaker Identification API using RESTful API calls. An audio file can be sent to the Speaker Identification API either through file upload from the web interface or the audio can be directly recorded through the web browser for both the enrollment and identification phases. The Speaker Identification API processes the audio and returns JSON data with a list of best matched profiles ranked in the order of highest similarity score for all the enrolled profiles. The profile with the highest similarity score is identified as the speaker in the audio file and it is printed onto the front-end interface.

5.4.1 Front End

The front-end consists of buttons to create an enrollment profile and identify speakers via live recording through the web browser or through a file upload. The latest live recording or the last uploaded file can be played through a built-in web media player that appears on the front-end interface when an audio signal is recorded, or an audio file is uploaded. The audio file in the web media player updates every time a new file is uploaded, or new audio is recorded. This is beneficial for users of the identification system so they can verify the audio that is currently being processed. The JSON data returned from the Speaker Identification API is printed directly onto the web interface in the order of execution so the users can see the real time execution of their requests. The returned JSON data from enrolling a profile is shown in Figure 5.4.1.1. The Speaker Identification API first returns some JSON data while creating a profile then some more data is returned and printed on the web interface during the enrollment phase. Next, it checks for the enrollment status and if the enrollment is complete then the user will be prompted to enter the name of the speaker for the enrollment audio so the profile id, which is a long string of random letters and numbers, can be mapped to a human identifiable name. This user prompt is shown in Figure 5.4.1.2. Upon completion of this prompt, a unique profile id will be mapped to the given user input. Figure 5.4.1.3 shows the results of an identification request. There were two enrolled profiles in this sample so each of the two profiles is assigned a similarity score and the profile with the highest similarity score is ultimately identified as the speaker in the recognition audio file.

```
Filename: 'BradAmy13mins-600-720secTrimSpeaker1Enrollment.wav'
(0.36MB)
creating profile
{
  "remainingEnrollmentsSpeechLength": 20,
  "locale": "en-us",
  "createdDateTime": "2021-04-08T14:17:51.343Z",
  "enrollmentStatus": "Enrolling",
  "modelVersion": null,
  "profileId": "c8832037-c21b-4b80-8fe5-6d6b7c696a88",
  "lastUpdatedDateTime": null,
  "enrollmentsCount": 0,
  "enrollmentsLength": 0,
  "enrollmentsSpeechLength": 0
}
enrolling
{
  "remainingEnrollmentsSpeechLength": 0,
  "profileId": "c8832037-c21b-4b80-8fe5-6d6b7c696a88",
  "enrollmentStatus": "Enrolled",
  "enrollmentsCount": 1,
  "enrollmentsLength": 24,
  "enrollmentsSpeechLength": 21.5,
  "audioLength": 24,
  "audioSpeechLength": 21.5
}
getting status
{
  "remainingEnrollmentsSpeechLength": 0,
  "locale": "en-us",
  "createdDateTime": "2021-04-08T14:17:51.343Z",
  "enrollmentStatus": "Enrolled",
  "modelVersion": "2019-11-01",
  "profileId": "c8832037-c21b-4b80-8fe5-6d6b7c696a88",
  "lastUpdatedDateTime": "04/08/2021 14:17:53",
  "enrollmentsCount": 1,
  "enrollmentsLength": 24,
  "enrollmentsSpeechLength": 21.5
}
enrollment complete
c8832037-c21b-4b80-8fe5-6d6b7c696a88 is now mapped to Speaker 1
```


Figure 5.4.1.1 Enrollment JSON Data

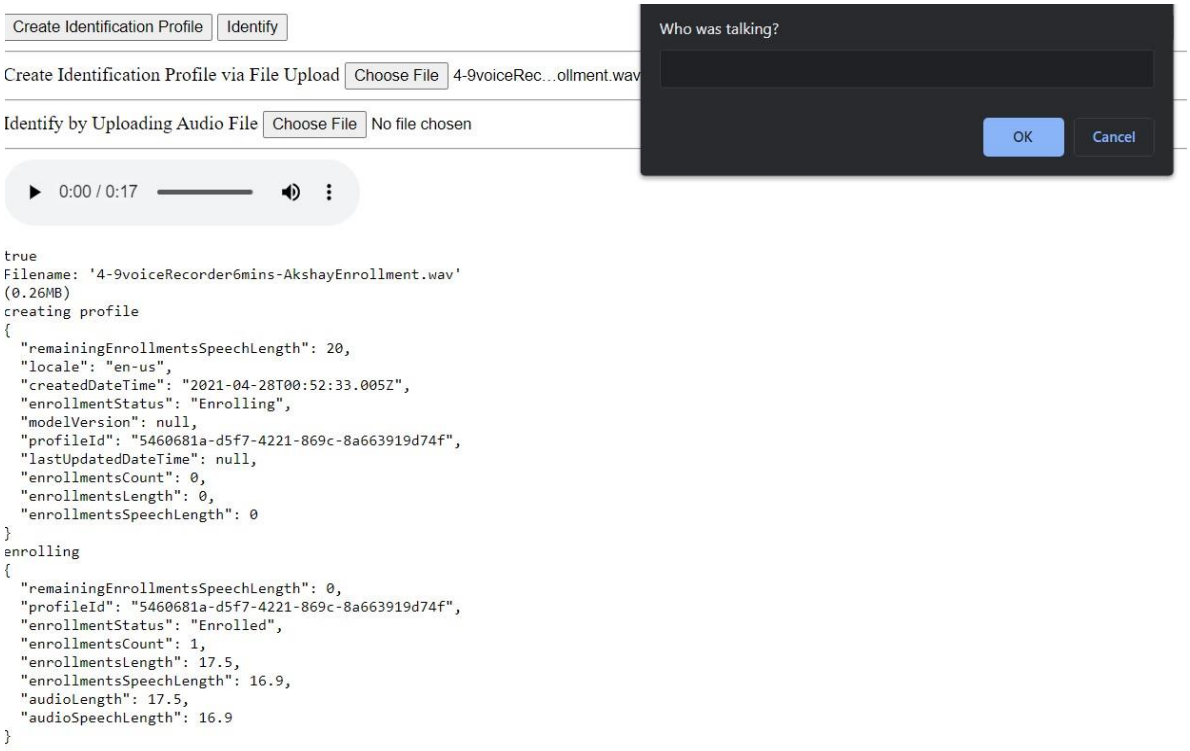

Figure 5.4.1.2 Front End User Prompt

```
Filename: 'BradAmy13mins-600-720secTrim3rd15sec.wav'
(0.22MB)
identifying profile
{
  "identifiedProfile": {
    "profileId": "c8832037-c21b-4b80-8fe5-6d6b7c696a88",
    "score": 0.8467979
  },
  "profilesRanking": [
    {
      "profileId": "c8832037-c21b-4b80-8fe5-6d6b7c696a88",
      "score": 0.8467979
    },
    {
      "profileId": "254472b8-46f2-4be5-a03b-3647f76c4361",
      "score": 0.19281173
    }
  ]
}
I think Speaker 1 was talking
```

Figure 5.4.1.3 Sample Identification Results

5.4.2 Back End

The initial process for setting up the back-end API configurations includes obtaining the necessary API endpoints from Microsoft Azure Cognitive Services. This is achieved through setting up a Cognitive Service resource through the Azure portal where the developer is given a key and an endpoint for authenticating the application. The key and endpoint must be private, and a new key can be generated upon request. There are two keys so one can be used when another one is in the regeneration process.

There are four required request URL endpoints to start identifying speakers. The required endpoints are create identification profile endpoint, create enrollment identification profile endpoint, enroll identification profile status endpoint, and identify profile endpoint. All these endpoints are essential for a complete Speaker Identification system.

Endpoint Name	Request URL
Create Identification Profile	https://{endpoint}/speaker/identification/v2.0/text-independent/profiles
Enroll Identification Profile	https://{endpoint}/speaker/identification/v2.0/text-independent/profiles/${profileId}/enrollments?ignoreMinLength=true
Enroll Identification Profile Status	https://{endpoint}/speaker/identification/v2.0/text-independent/profiles/${profileId}
Identify Profile	https://{endpoint}/speaker/identification/v2.0/text-independent/prfiles/identifySingleSpeaker?profileIds=${Ids}&ignoreMinLenght=true

Table 5.4.2 API endpoints and request URLs

Table 5.4.2 contains all the request URLS that were used in the Speaker Identification system. The{endpoint} section of the request URL is ‘westus.api.cognitive.microsoft.com’ because Speaker Recognition is still in the development phase and it is currently only offered in the US-West region.

There are 3 JavaScript files that handle the entire enrollment-identification process in the Speaker Identification system. The identification.js file contains all the endpoints and the functions to create a profile, enroll a profile, poll for enrollment status, and send audio to the identification endpoint. The core.js file contains the functions to add the recorded or uploaded audio to the web interface so the user can listen to it through a web media player. The core.js file also contains the Speaker Recognition API configuration and functionality for cross-browser audio recording using the web audio API. Finally, the recorder.js file handles all the required configurations when processing an audio file in the web such as maintaining proper encodings as well as enabling the functionality to download the recorded audio from the web interface.

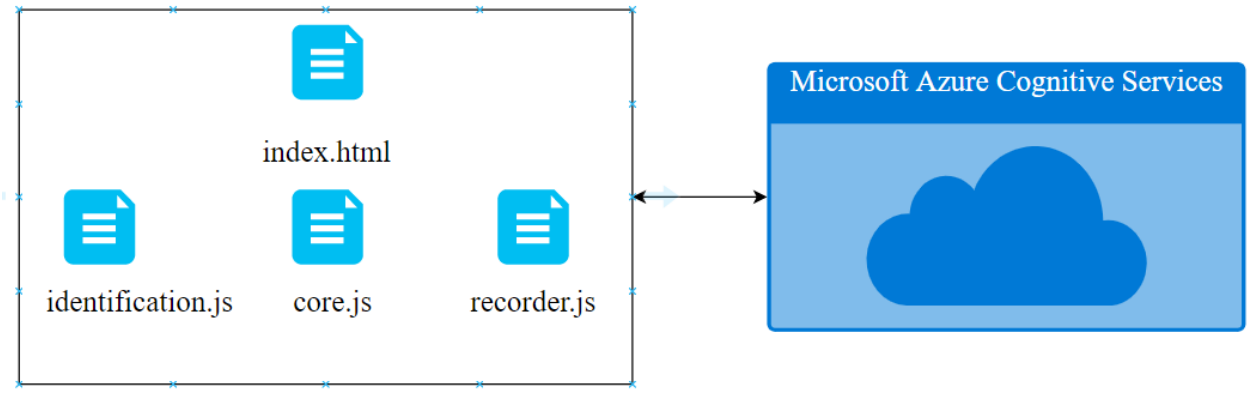

Figure 5.4.2 Speaker Identification Architecture

6. Evaluation

The collection and analysis of the data along with the results and the error rates are discussed in this section.

6.1 Dataset

Conversational data was obtained from the Synthetic Diarization Corpus (SDC) then processed using both Speaker Diarization and Identification. Data was also collected by our research team during their daily interactions using different voice recorders including a voice recorder Android application built for this research study by Youngbin Ha who is one of the research members on the autism research team. We emulated various realistic conversational scenarios with different types of speakers engaging in back-and-forth conversation while collecting the audio data.

Synthetic Diarization Corpus (SDC) is freely available for diarization research and it contains dialogs constructed from LibriSpeech Corpus. It offers over 90 hours of training data and over 9 hours of each test and development data. There was a choice of both two person and three person dialogues with and without overlap. Overlap data refers to situations where a speaker starts speaking without waiting for the previous speaker to finish speaking. Data from SDC includes both speaker and phoneme segmentation along with timing information in several formats but we only used the wav format which works best with our proposed methods.

Audio was also collected during daily interactions and meetings of our research team in one, two, and three-person conversational settings. This audio provides more realistic data that includes conversational speech with pauses, loss of audio quality and various other environmental effects such as the recording device receiving notifications, and noise caused due to movement of the recording device. Overall, this reduces the quality of the audio data which makes it more challenging to process the data with our proposed methods.

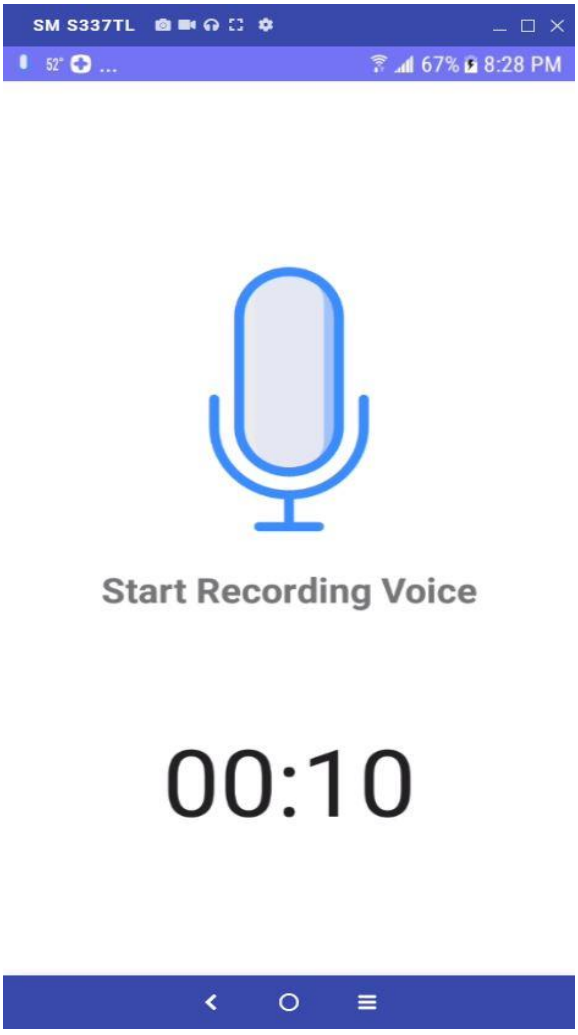

Figure 6.1 Voice Recorder Android App UI

6.2 Evaluation Process

The evaluation metric is obtained by calculating the total time spoken for each speaker from the Manual Validation results and that value is compared with the total time spoken calculation from both Speaker Diarization and Speaker Identification to determine the accuracy of the proposed methods. Manual Validation is done through listening to the audio file and determining which speaker spoke when. The first step in the data analysis phase is to preprocess the audio file by segmenting it into 2-minute-long (120 seconds) audio segments. The reason behind this choice is the time taken by the diarization algorithm to execute is directly proportional to the length of the audio file. There is also a high likelihood of the error propagating throughout the entire audio file if it is diarized in the same iteration. The process of analyzing the data begins with passing each of the 2-minute-long (120 seconds) audio segments to the Speaker Diarization algorithm then observing the audio file being segmented and clustered based on which speaker spoke when. The results from the Speaker Diarization algorithm are produced in tuples such as (speaker label, start time, end time). The total time spoken for each speaker is calculated from the Speaker Diarization algorithm results by calculating the sum of all the differences between the start and end time for each speaker. The next step is to manually validate each 2-minute-long (120 seconds) audio segment by listening to the audio file then determining the start and end times for each speaker based on when they spoke. The manual validation step is used as the ground truth and the results are noted in the same tuple format such as (speaker label, start time, end time) like the diarization results. The total time spoken for each speaker is also calculated from the manual validation results in a similar method used for the Speaker Diarization results. The results from this manual validation step will be later used as a comparison metric with the results from both

Speaker Diarization and Speaker Identification so we can establish a performance metric and determine the accuracy of our proposed methods. Next, the 2-minute-long (120 seconds) audio segments are further segmented into 15-second-long audio segments then each 15-second-long audio segment is sent to the Speaker Identification API through a RESTful API call. The reason behind this choice is to take a fine sample rate in an attempt to decrease the identification error rate. The Speaker Identification API returns JSON data which is printed onto the web interface in the order of execution. Our goal in this phase is to determine if a certain enrolled speaker is present in a certain 15-second-long segment. Each of the enrolled speakers is isolated in the enrollment process and compared against each of the 15-second-long audio segments until it encompasses the duration of the entire audio file. A speaker is identified in a segment when a similarity score of greater than the minimum acceptable threshold of 0.50 is achieved for the enrolled profile. Next all the enrolled profiles are enrolled at the same time then compared against each of the 15-second-long audio segments to compare the performance when there are one or many enrolled profiles. Finally, all the similarity scores of greater than 0.50 are added and multiplied by 15, which accounts for the length of 15-second-long audio segment, to estimate the total time spoken for that speaker. At the end of this process, we will have a total time spoken value for each speaker from each of the Speaker Diarization, Speaker Identification and Manual Validation methods. These results are further analyzed and compared to determine the performance, accuracy, and error rate of the proposed methods.

6.3 Results

In this section, the results from processing 4 audio files from the Synthetic Diarization Corpus (SDC) dataset containing three speakers, three speakers with overlap, two speakers, and two speakers with overlap are discussed. Results from processing 5 audio files that were collected during the initial pilot testing study are also discussed. The results from the audio data collected through our voice recorder Android application are also presented by processing various audio files containing speakers from both genders and speakers with similar voice characteristics.

The results from an audio file obtained from the Synthetic Diarization Corpus dataset with a length of 8 minutes and 25 seconds containing three speakers without any overlapped speech portions are shown in the tables 6.3.1 – 6.3.3 and figures 6.3.1 – 6.3.4. The data shows time in seconds of how long each speaker spoke during a certain audio segment (i.e., first 2 minutes, second 2 minutes, third 2 minutes, so on and so forth).

All the tables for the SDC dataset results have four columns – Audio Segments, Manual Validation, Speaker Diarization, and Speaker Identification. The first column refers to each of the 2-minute-long segments that the audio file is initially segmented into. The following three columns show the total time spoken in seconds during each audio segment for the three different methods being compared.

6.3.1 Dataset 1: SDC with Three Speakers without Overlap

Table 6.3.1.1 and Figure 6.3.1.1 show the results for Speaker 0 in the audio file obtained from SDC containing three speakers without any overlapped speech.

Audio Segments	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	42.9	37.74	43.05
Second 2 minutes	42	35.4	44.55
Third 2 minutes	42	32.46	31.5
Fourth 2 minutes	55	60.92	47.7
Last 25 seconds	12	10.5	9.8
Total 8 minutes and 25 seconds	193.9	177.02	176.6

Table 6.3.1.1 Three Speakers – Speaker 0

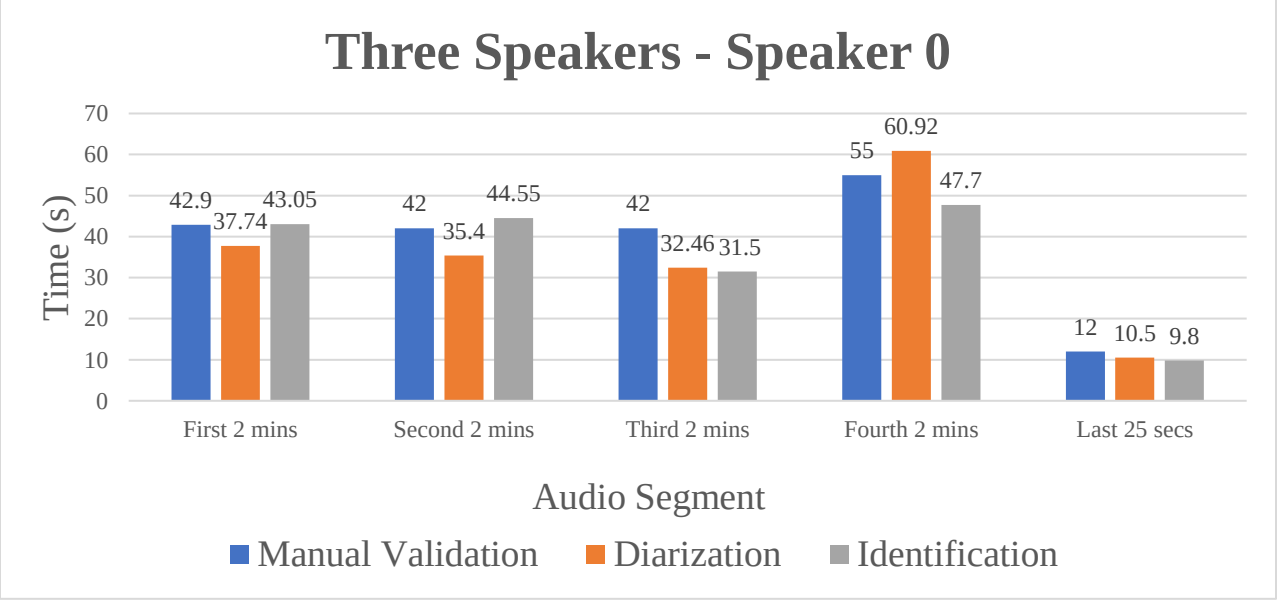

Figure 6.3.1.1 Three Speakers – Speaker 0 chart

The total time spoken for Speaker Diarization and Speaker Identification methods are lower than the Manual Validation results for Speaker 0. This is partly due to the Speaker Diarization algorithm using a VAD to trim the non-speech and silent portions from the audio file. Both Diarization and Identification methods yield similar results for the total time spoken calculation for Speaker 0.

Table 6.3.1.2 and Figure 6.3.1.2 show the results for Speaker 1 in this audio file obtained from SDC that contains three speakers without any overlapped speech.

Audio Segments	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	37.1	33.38	38.1
Second 2 minutes	40	41.6	42.6
Third 2 minutes	55	56.78	62.1
Fourth 2 minutes	46	37.14	36
Last 25 seconds	5	5.6	10.35
Total 8 minutes and 25 seconds	183.1	174.5	189.15

Table 6.3.1.2 Three Speakers – Speaker 1

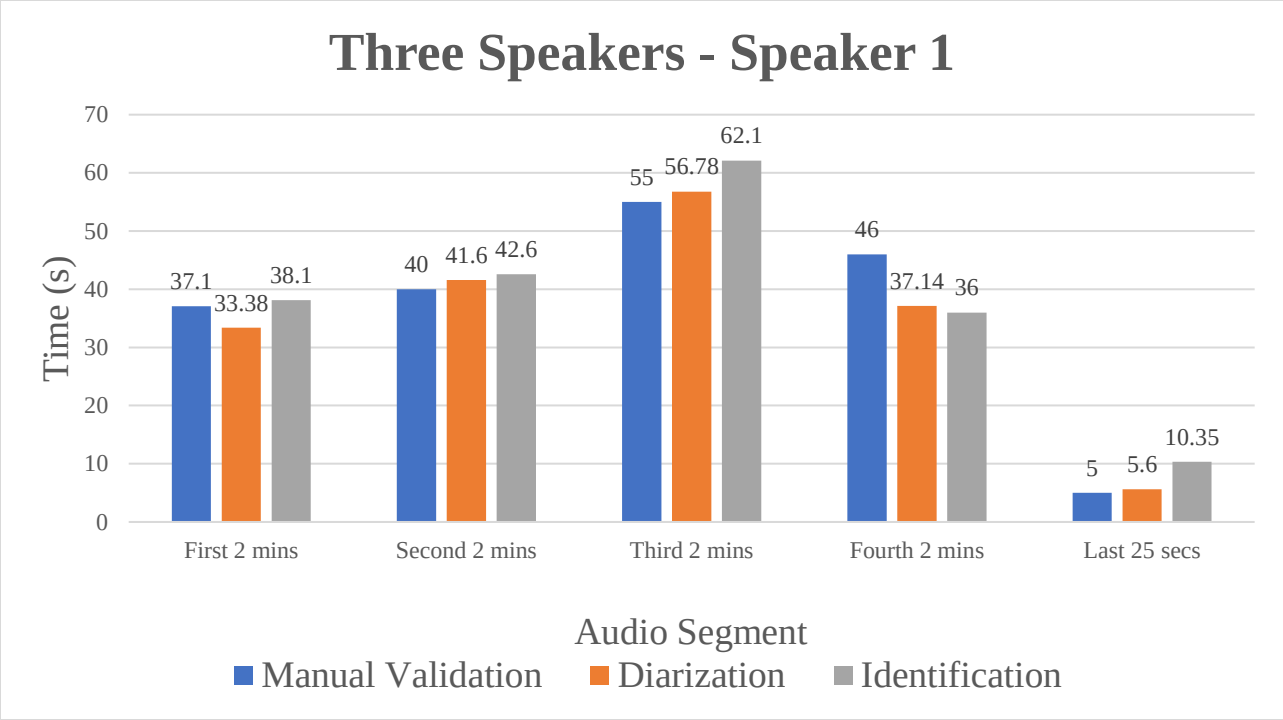

Figure 6.3.1.2 Three Speakers – Speaker 1 chart

Even though the total time spoken value calculated using Speaker Diarization is lower compared to the Manual Validation result, Speaker Identification yields a higher total time spoken value for Speaker 1. This is justified because Speaker Diarization trims out the non-speech portions of the audio file whereas Speaker Identification is an estimate based on how many 15-second-long segments and with what similarity score is Speaker 1 identified. Speaker 1 was identified with high similarity scores since there is no overlap in the audio signal and as result the total time spoken value calculated using the Speaker Identification method yields a higher value than both Manual Validation and Speaker Diarization.

Audio Segments	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	40	38.64	27
Second 2 minutes	36	31.68	27.75
Third 2 minutes	23	20.22	29.85
Fourth 2 minutes	19	15.48	17.4
Last 25 seconds	8	6.66	8.85
Total 8 minutes and 25 seconds	126	112.68	110.85

Table 6.3.1.3 Three Speakers – Speaker 2

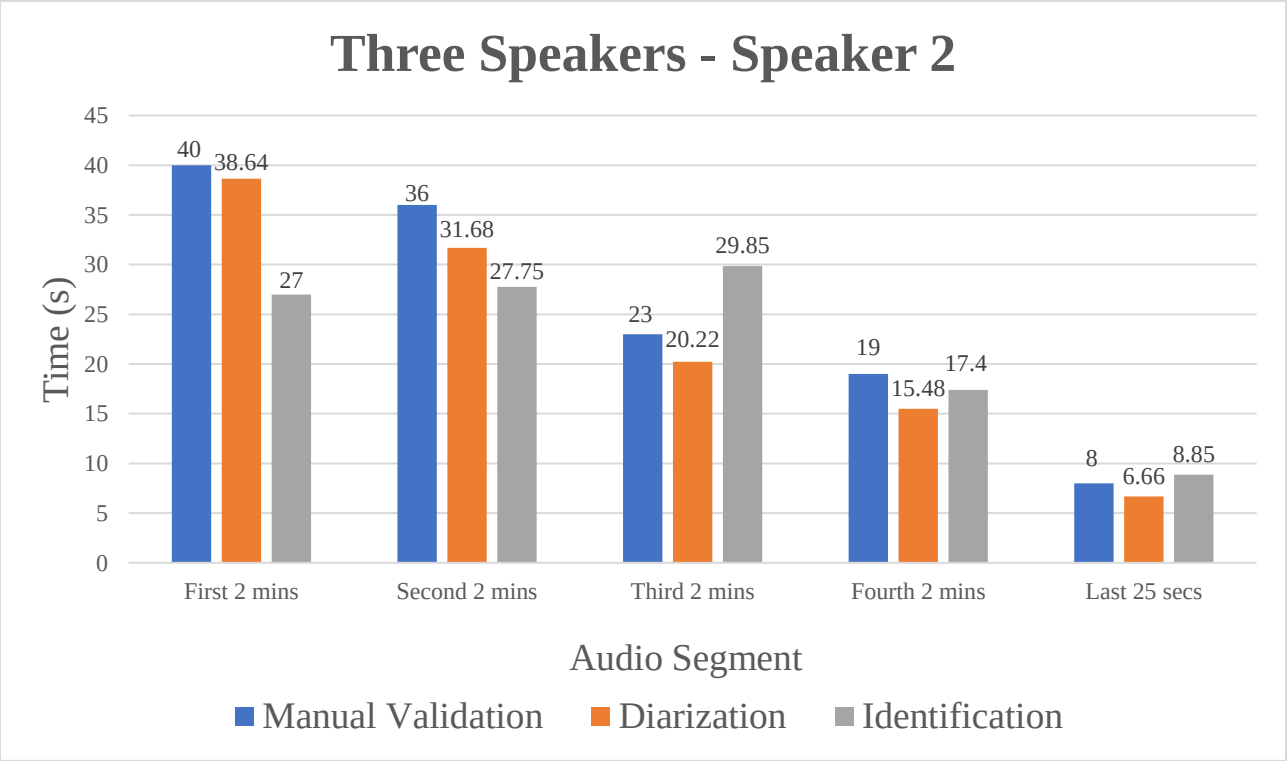

Figure 6.3.1.3 Three Speakers – Speakers 2 chart

Speaker 2 results are like Speaker 0 results because both the Diarization and Identification methods yield a lower total time spoken value compared to the Manual Validation results. Once again, Diarization results being lower can be justified due to the non-speech portions being trimmed out. The significant takeaway here is that both Diarization and Identification yield similar results.

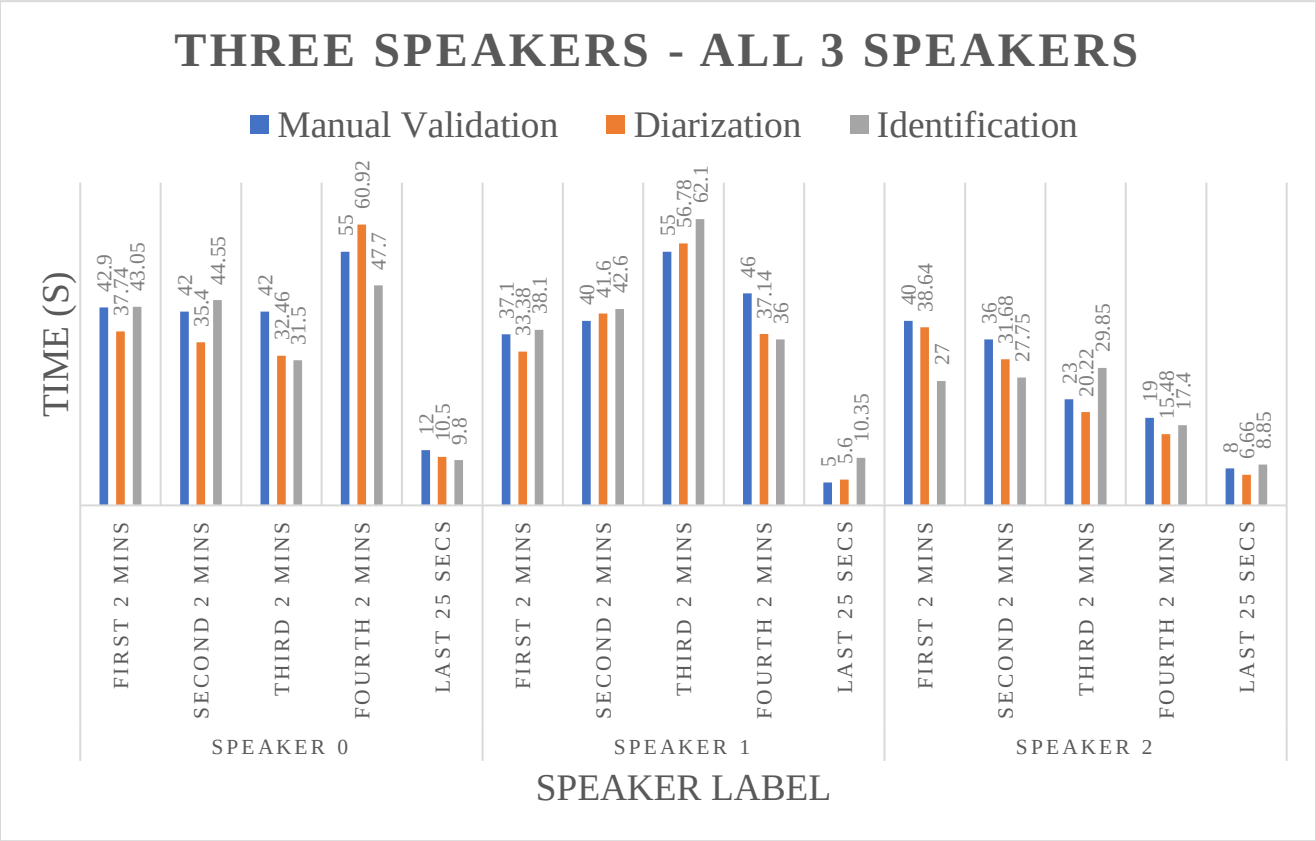

Figure 6.3.1.4 Three Speakers – All Three Speakers chart

Figure 6.3.1.4 shows the total time spoken splits for all three speakers in the audio file in the same chart. Furthermore, the data is segmented into 2-minute segments for each speaker until the entire length of the audio file is covered. Significant takeaway here is that both Diarization and Identification performed remarkably well.

6.3.2 Dataset 2: SDC with Three Speakers with Overlap

The results from an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset with a length of 6 minutes and 14 seconds containing three speakers with overlapped speech portions are shown in the tables 6.3.2.1 – 6.3.2.3 and figures 6.3.2.1 – 6.3.2.4. The data shows time in seconds of how long each speaker speaks during a certain audio segment.

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	44	40.56	48
Second 2 minutes	31	28.08	21.3
Third 2 minutes and 14 seconds	69	69.06	73.05
Total 6 minutes and 14 seconds	144	137.7	142.35

Table 6.3.2.1 Three Speakers with Overlap – Speaker 0

The total time spoken calculated through Manual Validation for Speaker 0 yields a value of 144 seconds compared to values of 137.7 seconds and 142.35 seconds for Speaker Diarization and

Speaker Identification, respectively. Overall, the total time spoken values for Speaker 0 for all three methods are close in proximity.

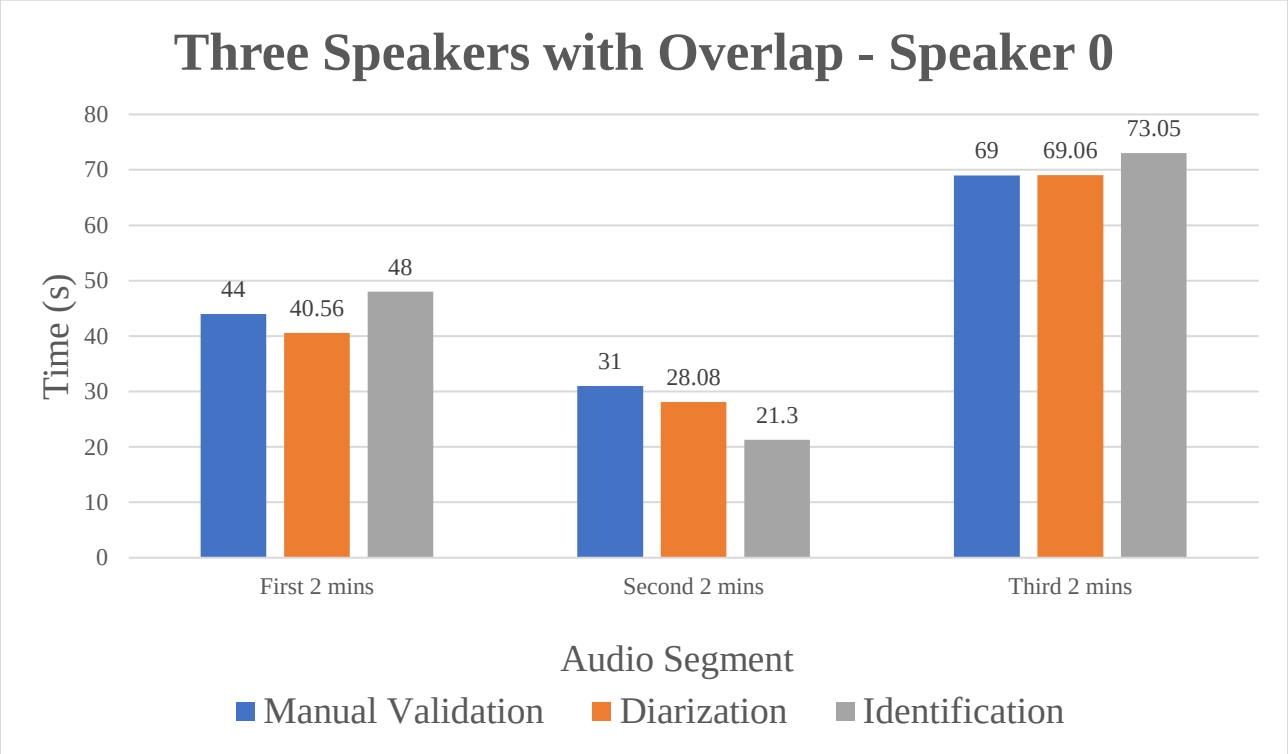

Figure 6.3.2.1 Three Speakers with Overlap – Speaker 0 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	25	25.5	33
Second 2 minutes	48	45.12	65.4
Third 2 minutes and 14 seconds	39	30.9	21.15
Total 6 minutes and 14 seconds	112	101.52	119.5

Table 6.3.2.2 Three Speakers with Overlap – Speaker 1

The total time spoken calculated through Manual Validation for Speaker 1 yields a value of 112 seconds which is close to the values of 101.52 seconds and 119.5 seconds for Diarization and Identification methods. Diarization values are lower than Manual Validation due to the silences being trimmed from the audio file. Identification values are greater than Manual Validation values due to the high similarity scores that Speaker 1 was identified with.

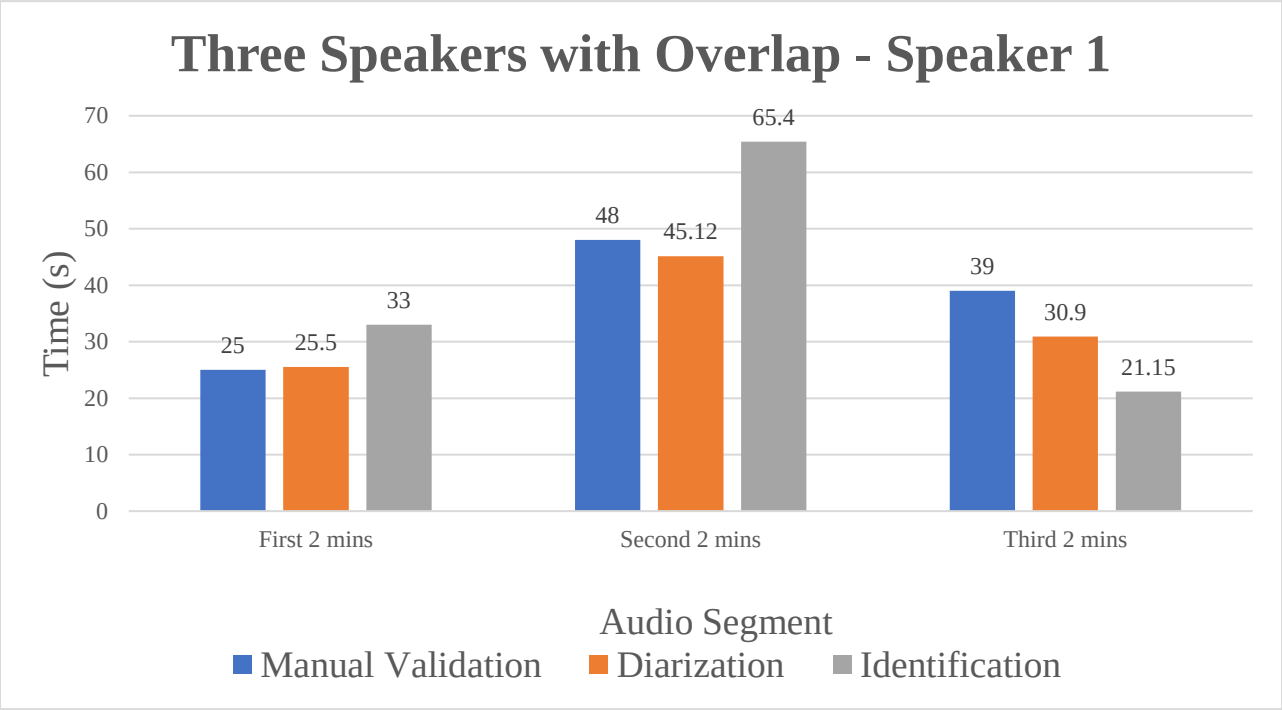

Figure 6.3.2.2 Three Speakers with Overlap – Speaker 1 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	51	45.8	36.75
Second 2 minutes	41	39.66	40.05
Third 2 minutes and 14 seconds	26	26.66	37.62
Total 6 minutes and 14 seconds	118	112.12	114.42

Table 6.3.2.3 Three Speakers with Overlap – Speaker 2

The total time spoken calculated through Manual Validation for Speaker 2 yields a value of 118 seconds compared to values of 112.12 seconds and 114.42 seconds for Diarization and Identification, respectively. Diarization and Identification results are especially close with a difference of about two seconds. The results from the Diarization method show the best performance in the third 2 minutes segment for Speaker 2. The results from the Identification method show the best performance in the second 2 minutes segment for Speaker 2. The first 2 minutes audio segment had the worst performance for both Diarization and Identification but the performance for both methods improved significantly in the following audio segments for Speaker 2 in this audio file.

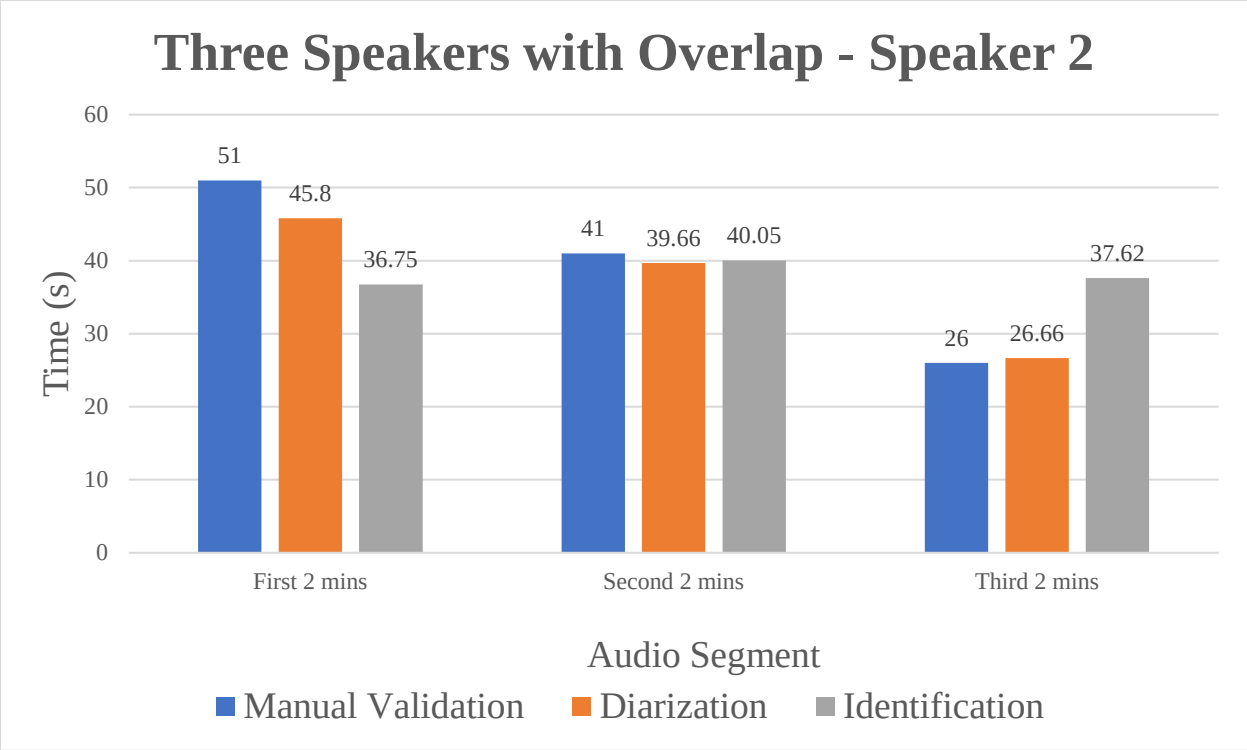

Figure 6.3.2.3 Three Speakers with Overlap – Speaker 2 chart

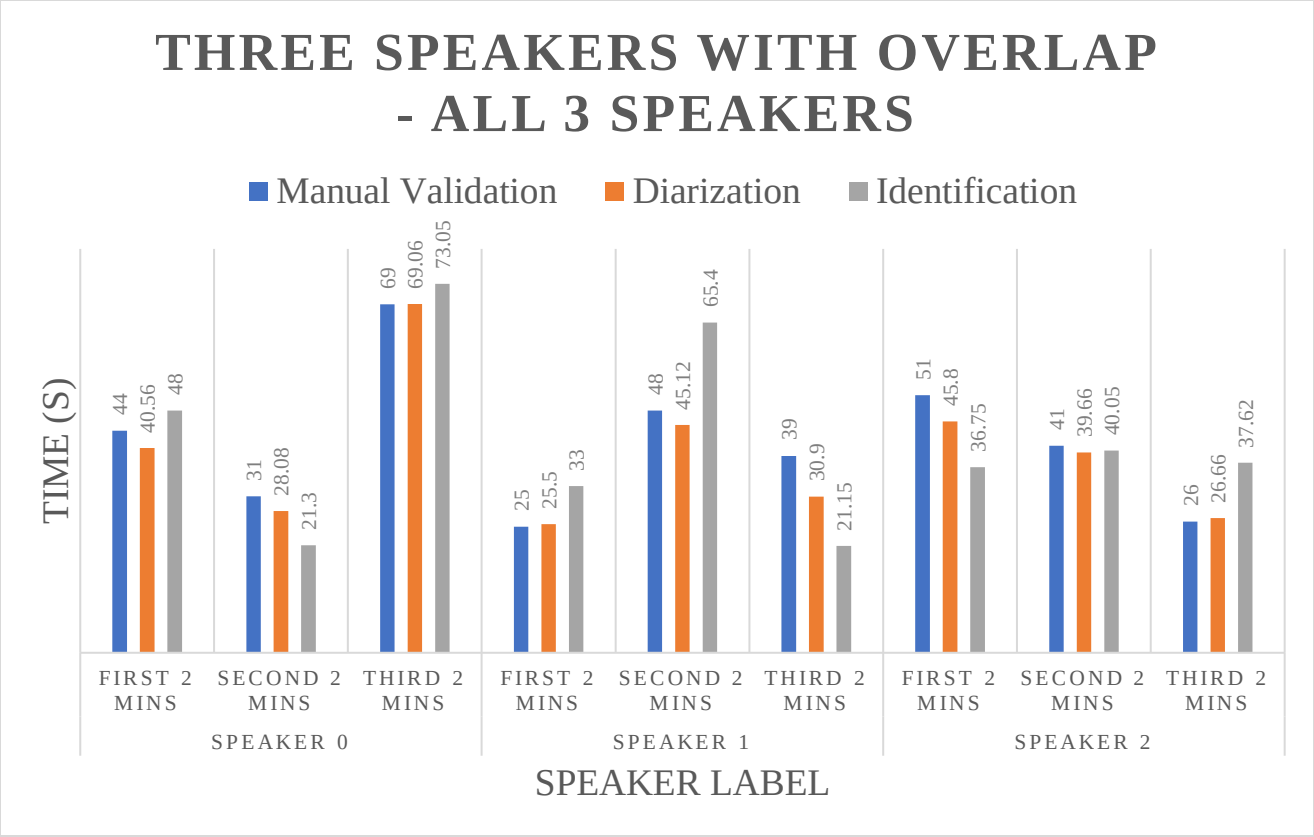

Figure 6.3.2.4 Three Speakers with Overlap – All Three Speakers chart

The total time spoken splits during the different 2-minute-long audio segments are shown for all three speakers in a single chart in Figure 6.3.2.4. A pattern that can be observed from the three speakers with overlap audio file results is that total time spoken calculated using the Speaker Identification method is greater than Diarization results for all three speakers. Diarization results are relatively close to Manual Validation results and follow a general trend of being less than both Identification and Manual Validation results. The results from this six-minute-long audio file with three speakers with overlapped speech further supports our initial expectation of the performance for both Diarization and Identification.

6.3.3 Dataset 3: SDC with Two Speakers without Overlap

The results from processing an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset with a length of 4 minutes and 53 seconds containing two speakers without any overlapped speech portions are shown in the tables 6.3.3.1 – 6.3.3.2 and figures 6.3.3.1 – 6.3.3.3. The data shows time in seconds of how long each speaker spoke during a certain audio segment.

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	64	79.08	51.3
Second 2 minutes	64	63.24	66.6
Last 53 seconds	22	17.24	9.15
Total 4 minutes and 53 seconds	150	159.56	127.05

Table 6.3.3.1 Two Speakers – Speaker 0

The total time spoken calculated using Manual Validation for Speaker 0 yields a value of 150 seconds compared to values of 159.56 seconds and 127.05 seconds for Diarization and Identification, respectively. There is only a nine second difference when comparing the Diarization results to Manual Validation results. This is a rare case where the Diarization values are greater than Manual Validation. Generally, Diarization values are less than Manual Validation results because the VAD in the Diarization algorithms trims out all the non-speech portions of the audio file. Identification performed worse than Diarization for Speaker 0 in this audio file.

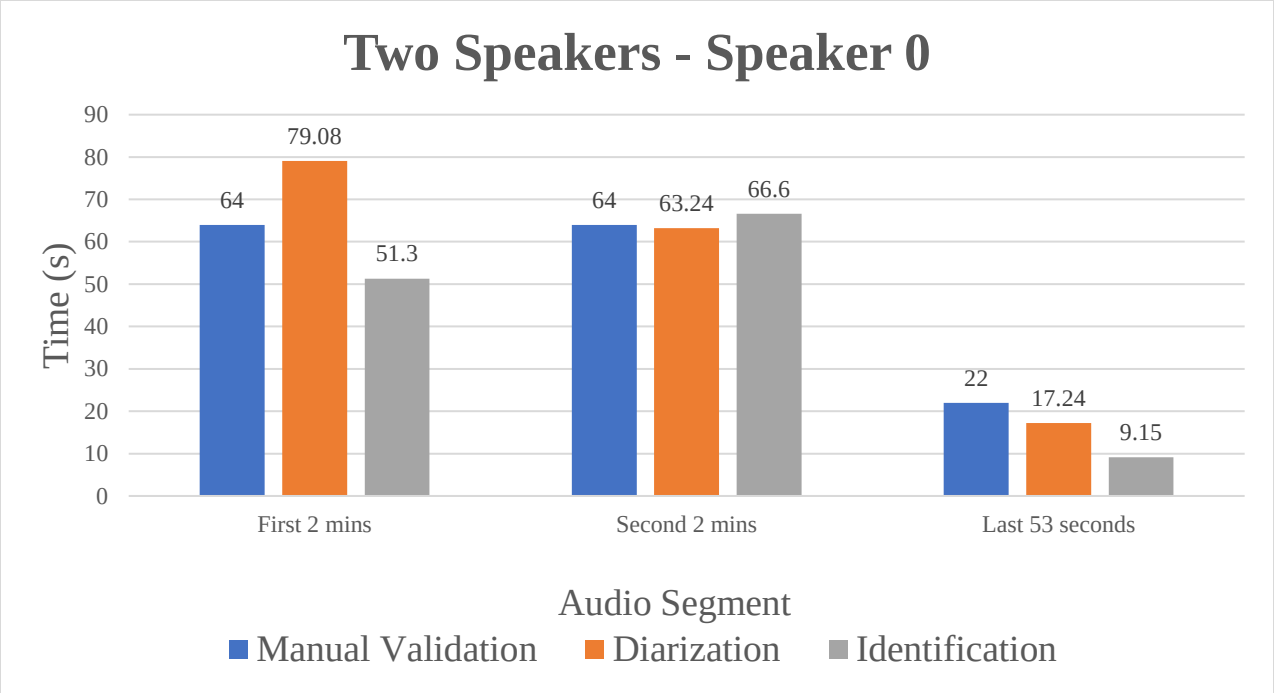

Figure 6.3.3.1 Two Speakers – Speaker 0 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	56	34.28	33
Second 2 minutes	56	51.02	23.1
Last 53 seconds	31	33.42	37.57
Total 4 minutes and 53 seconds	143	118.72	93.67

Table 6.3.3.2 Two Speakers – Speaker 1

The total time spoken calculated through Manual Validation for Speaker 1 yields a value of 143 seconds compared to values of 118.72 seconds and 93.67 seconds for Diarization and Identification, respectively. Diarization is back to the trend of being less than the Manual

Validation results. Identification performance was poor for Speaker 1 in this audio file with a difference of approximately fifty seconds when compared to the Manual Validation results. This is primarily caused due to the discrepancy in the second 2 minutes audio segment. The results are visualized in Figure 6.3.3.2.

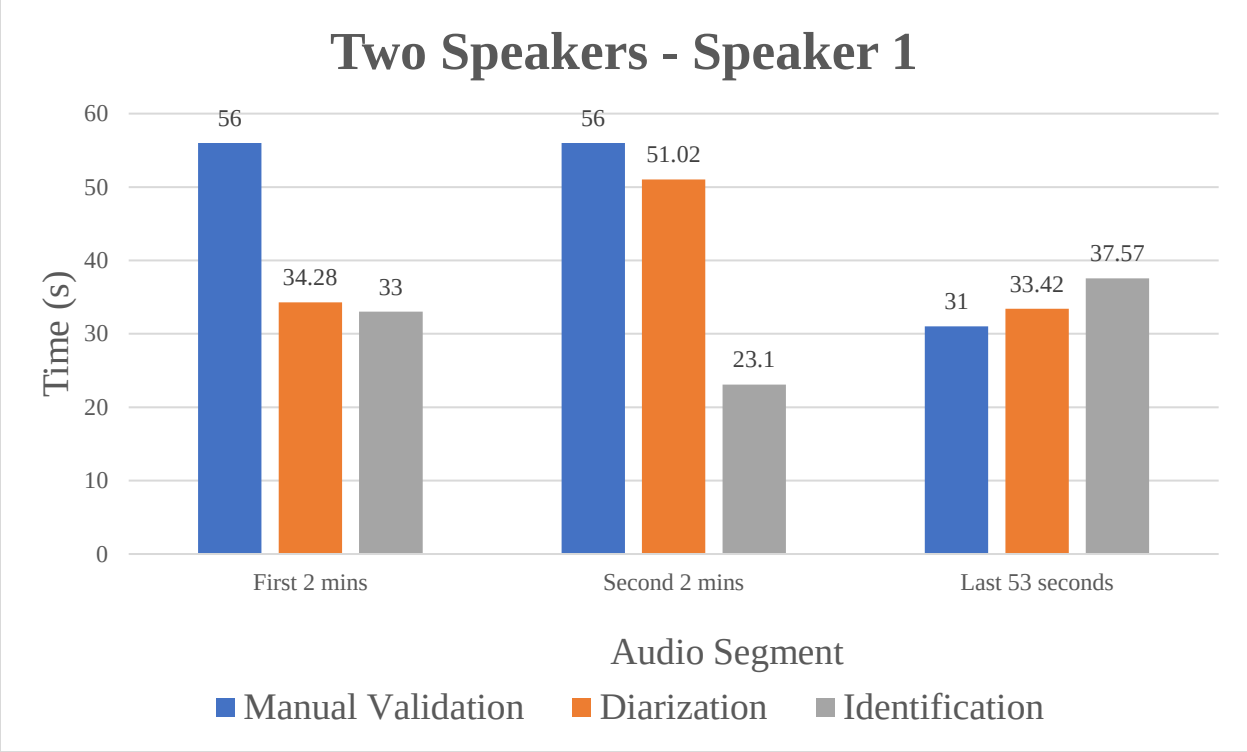

Figure 6.3.3.2 Two Speakers – Speaker 1 chart

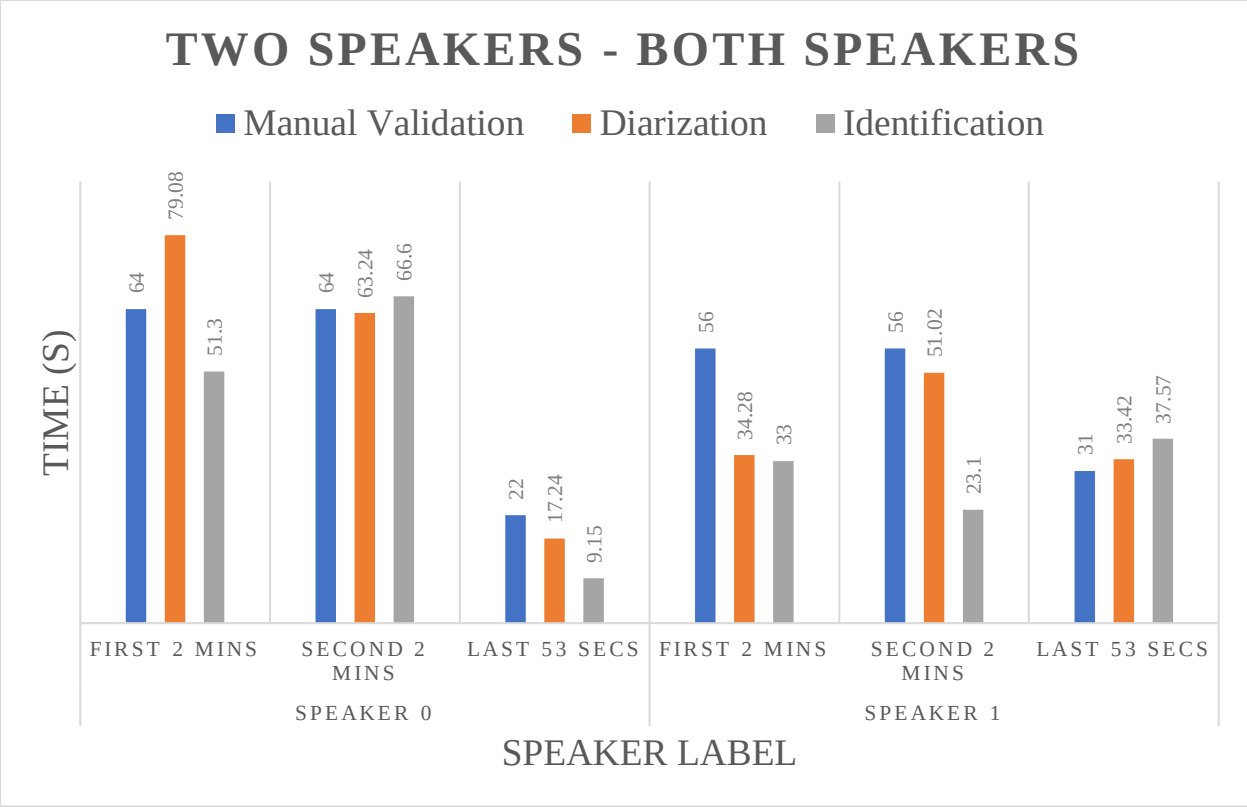

Figure 6.3.3.3 Two Speakers – Both Speakers chart

The total time spoken splits over the 2-minute-long audio segments for both speakers in an audio file with two speakers without any overlapped speech portions is shown in the Figure 6.3.3.3. Speaker Diarization performed better than Speaker Identification for this audio file. Speaker Diarization results were a lot closer to Manual Validation results than results from the Speaker Identification method.

6.3.4 Dataset 4: SDC with Two Speakers with Overlap

The results from an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset with a length of 6 minutes and 42 seconds containing two speakers with overlapped speech portions are shown in the tables 6.3.4.1 – 6.3.4.2 and figures 6.3.4.1 – 6.3.4.3 below. The data shows time in seconds of how long each speaker spoke during a certain audio segment.

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	70	68.34	77.25
Second 2 minutes	45	40.62	32.25
Third 2 minutes	84	83.54	86.85
Last 42 seconds	14	9.56	11.55
Total 6 minutes and 42 seconds	213	202.06	207.9

Table 6.3.4.1 Two Speakers with Overlap – Speaker 0

The total time spoken calculated through Manual Validation for Speaker 0 yields a value of 213 seconds compared to values of 202.06 seconds and 207.9 seconds for Diarization and Identification, respectively. The difference is minimal between both Diarization and Identification results compared to the Manual Validation results. Diarization generally tends to be less than Manual Validation. Identification results are closer to Manual Validation results with a difference of only five seconds. These results can be visualized in Figure 6.3.4.1.

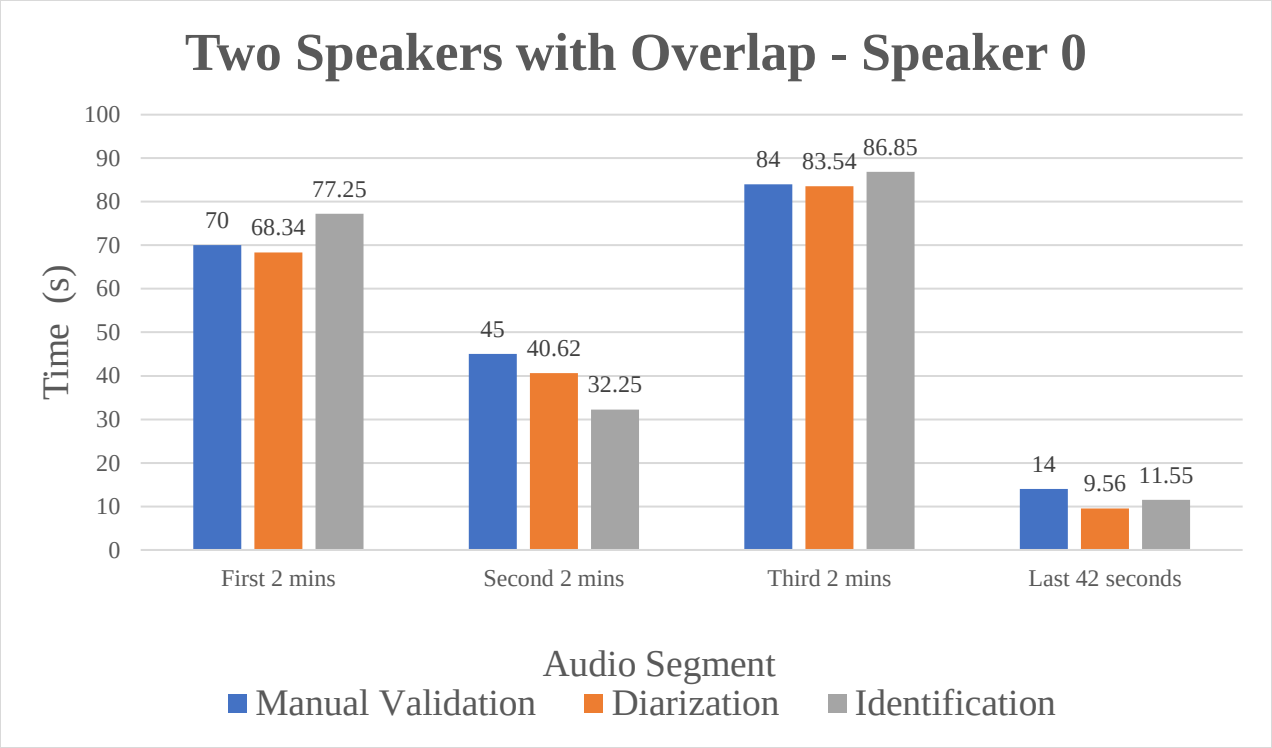

Figure 6.3.4.1 Two Speakers with Overlap – Speaker 0 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	50	47.42	20.85
Second 2 minutes	75	75.86	57.45
Third 2 minutes	36	29.76	24.15
Last 42 seconds	28	29.16	24.03
Total 6 minutes and 42 seconds	189	182.2	126.48

Table 6.3.4.2 Two Speakers with Overlap – Speaker 1

The total time spoken calculated through Manual Validation yields a value of 189 seconds for Speaker 1 compared to values of 182.2 seconds and 126.48 seconds for Diarization and

Identification, respectively. Diarization performed well for Speaker 1 with a total time spoken value difference of less than seven seconds when compared to Manual Validation results. Identification performed worse for Speaker 1 than Speaker 0 for this audio file. The main discrepancy behind Identification's poor performance was introduced in the first two minutes audio segment and was propagated through the total time spoken results for this entire audio file. These results can be visualized in figure 6.3.4.2.

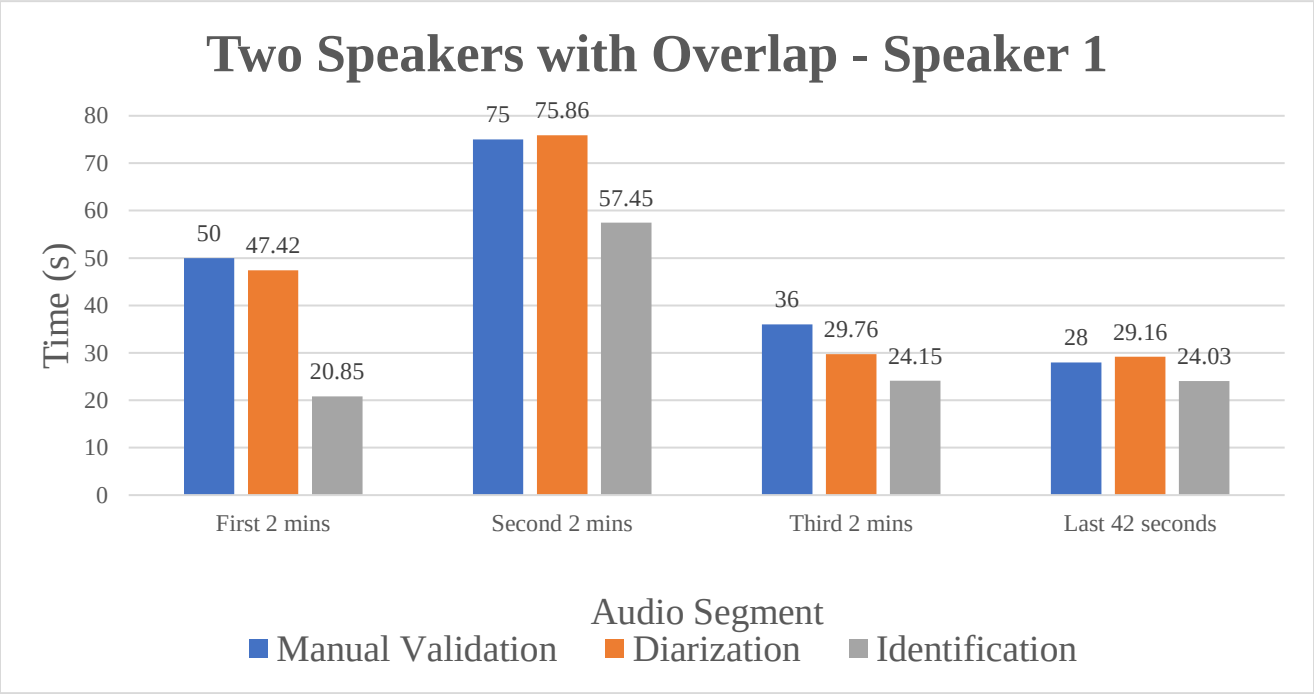

Figure 6.3.4.2 Two Speakers with Overlap – Speaker 1 chart

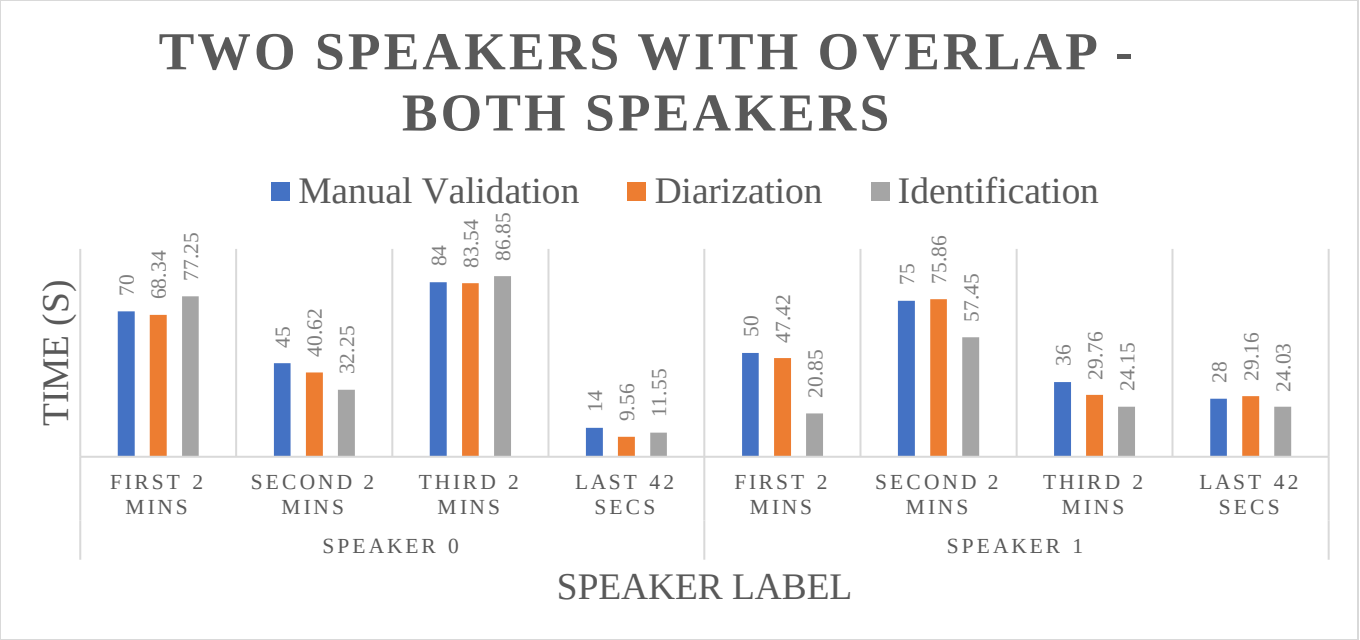

Figure 6.3.4.3 Two Speakers with Overlap – Both Speakers chart

The total time spoken splits over the different 2-minute-long audio segments for both speakers in the audio file with two speakers with overlap is shown in the Figure 6.3.4.3. Both Speaker Diarization and Speaker Identification results were close to Manual Validation results for both speakers in this audio file. However, Speaker Identification did not perform as well for speaker 1 as it did for speaker 0 largely due to the disparity in the results for the first two minutes audio segment.

6.3.5 Dataset 5: Three Female Speakers Conversation

Our research team collected three different audio files that are roughly one minute in length. One file contains three female speakers, another file contains two female speakers and finally the last file has only one female speaker. All the speakers in these audio files have similar voice characteristics and they emulate a realistic conversational exchange with a lot of overlap in speech. The speakers were also wearing masks while collecting this audio data to comply with COVID-19 pandemic safety guidelines which may affect the quality of the data and ultimately affect the performance of the proposed methods. This data was used in the initial pilot testing study and the results from this data can be observed in tables 6.3.5 - 6.3.7 and figures 6.3.5 – 6.3.7. The tables show total time spoken for each speaker for Manual Validation, Speaker Diarization and Speaker Identification. The figures visualize the data from these tables.

Speaker	Manual Validation	Speaker Diarization	Speaker Identification
Speaker 0	18	18.8	38.62
Speaker 1	27	27	35.65
Speaker 2	11	8.85	13.65
All 3 Speakers	56	54.65	87.92

Table 6.3.5 Three Female Speakers Conversation

Diarization performed well for all three speakers in this audio file. Identification performed worse than Diarization for this audio file. The identification values were higher than Manual Validation results due to speakers being identified with a high similarity score.

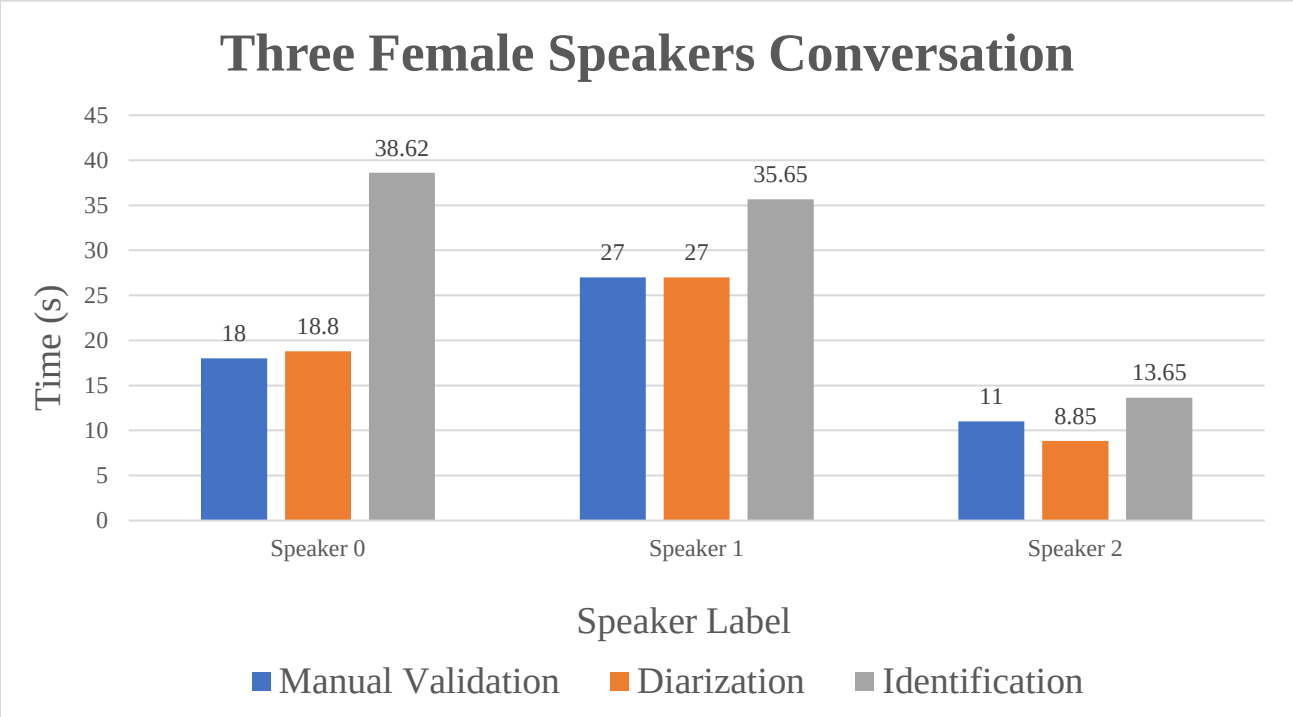

Figure 6.3.5 Three Female Speakers Conversation chart

6.3.6 Dataset 6: Two Female Speakers Conversation

Speaker	Manual Validation	Speaker Diarization	Speaker Identification
Speaker 0	24	27.98	23.82
Speaker 1	22	15.48	14.85
Both Speakers	46	43.46	38.67

Table 6.3.6 Two Female Speakers Conversation

The results for an audio file with two female speakers with similar voice characteristics can be seen in Table 6.3.6 and Figure 6.3.6. This file was recorded in the same environment as the previous one with three female speakers with similar voice characteristics. Overall, Speaker 0 had better performance for both Diarization and Identification in this audio file with two female speakers.

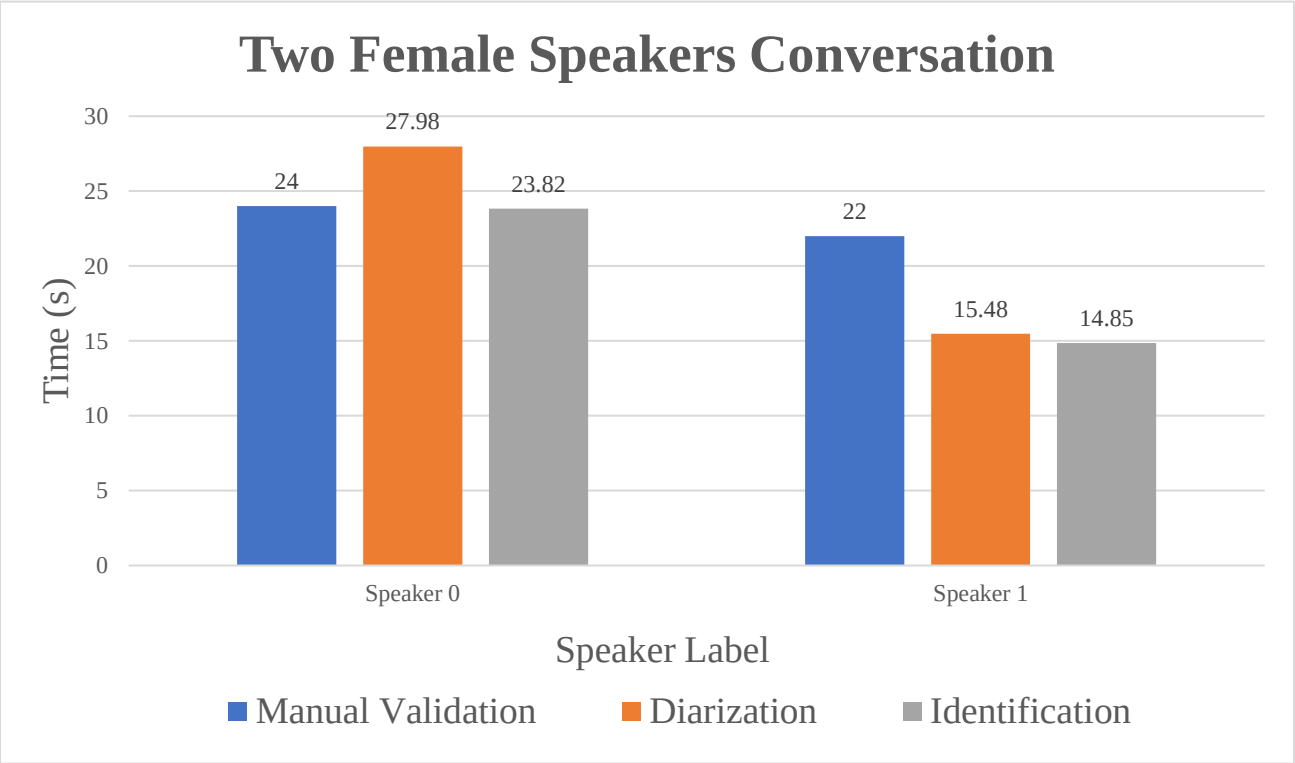

Figure 6.3.6 Two Female Speakers Conversation chart

6.3.7: Dataset 7: One Female Speaker Monologue

Speaker	Manual Validation	Speaker Diarization	Speaker Identification
Speaker 0	57	55.04	51.75

Table 6.3.7 One Female Speaker Monologue

The results for an audio file with one female speaker are shown in Table 6.3.7 and Figure 6.3.7.

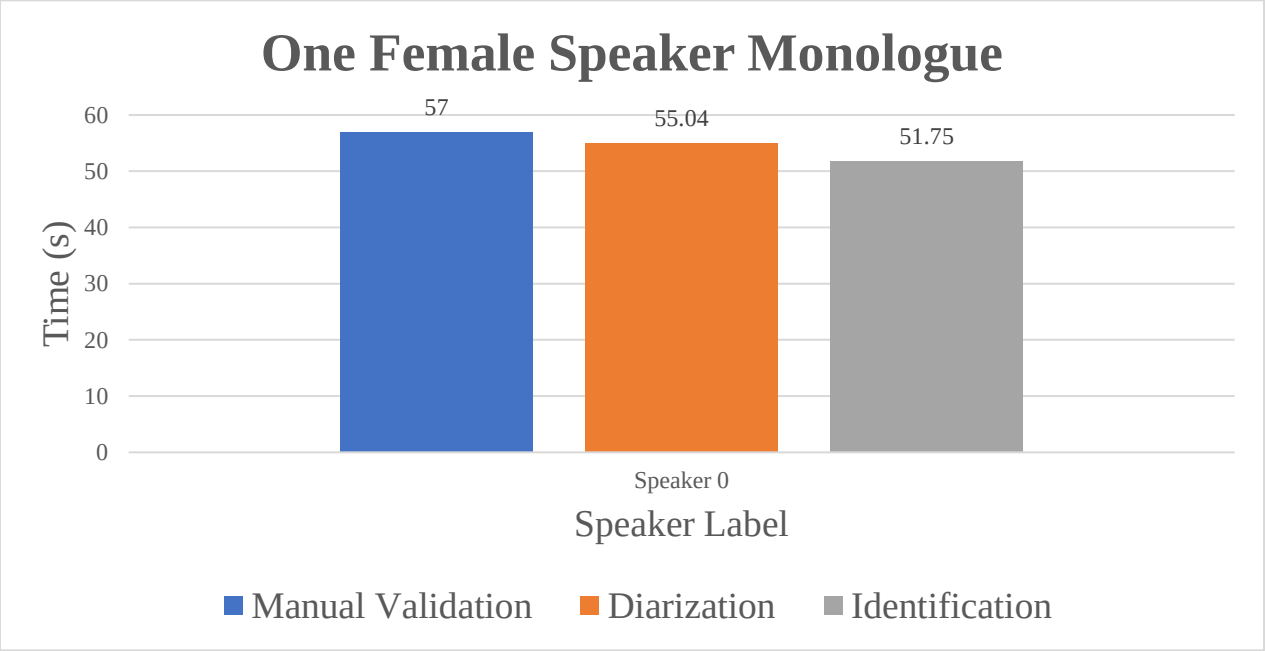

Figure 6.3.7 One Female Speaker Monologue chart

6.3.8 Dataset 8: One Male and One Female Conversation

The results from a 13-minute-long audio file containing one male and one female speaker engaging in back-and-forth conversation are shown in tables 6.3.8.1 – 6.3.8.2 and figures 6.3.8.1 – 6.3.8.3. This audio file is recorded with the voice recorder Android application that was developed for this study. This emulates various realistic challenges that arise such as the target speaker being closer to the recording device which affects the audio quality of the other speakers. There are multiple pauses and a lot of silences throughout the exchange which is common in realistic conversations. There is also noise due to the recording device moving and receiving notifications throughout the recording.

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	26	15.48	0
Second 2 minutes	18	56	7.35
Third 2 minutes	52	15.48	34.95
Fourth 2 minutes	30	11.1	19.35
Fifth 2 minutes	25	57.06	0
Sixth 2 minutes	49	37.14	27.9
Last 1 minute and 23 seconds	36	20.28	17.85
Total 13 minutes and 23 seconds	236	212.54	107.4

Table 6.3.8.1 One Male and One Female Conversation – Speaker 0

The total time spoken calculation through Manual Validation for Speaker 0 in this audio file yields a result of 236 seconds compared to values of 212.54 seconds and 107.4 seconds for Diarization and Identification, respectively. Diarization performed well with only a difference of 24 seconds while Identification performed poorly with a significant difference when compared to Manual Validation results. The reason behind the poor performance was due to bad audio quality where Speaker 0 was barely audible through certain segments of the audio file and as a result, Speaker 0 was not identified at all for that segment. This is evident in the first two minutes and fifth two minutes audio segments where the total time spoken for that segment was 0 seconds for the Speaker Identification method.

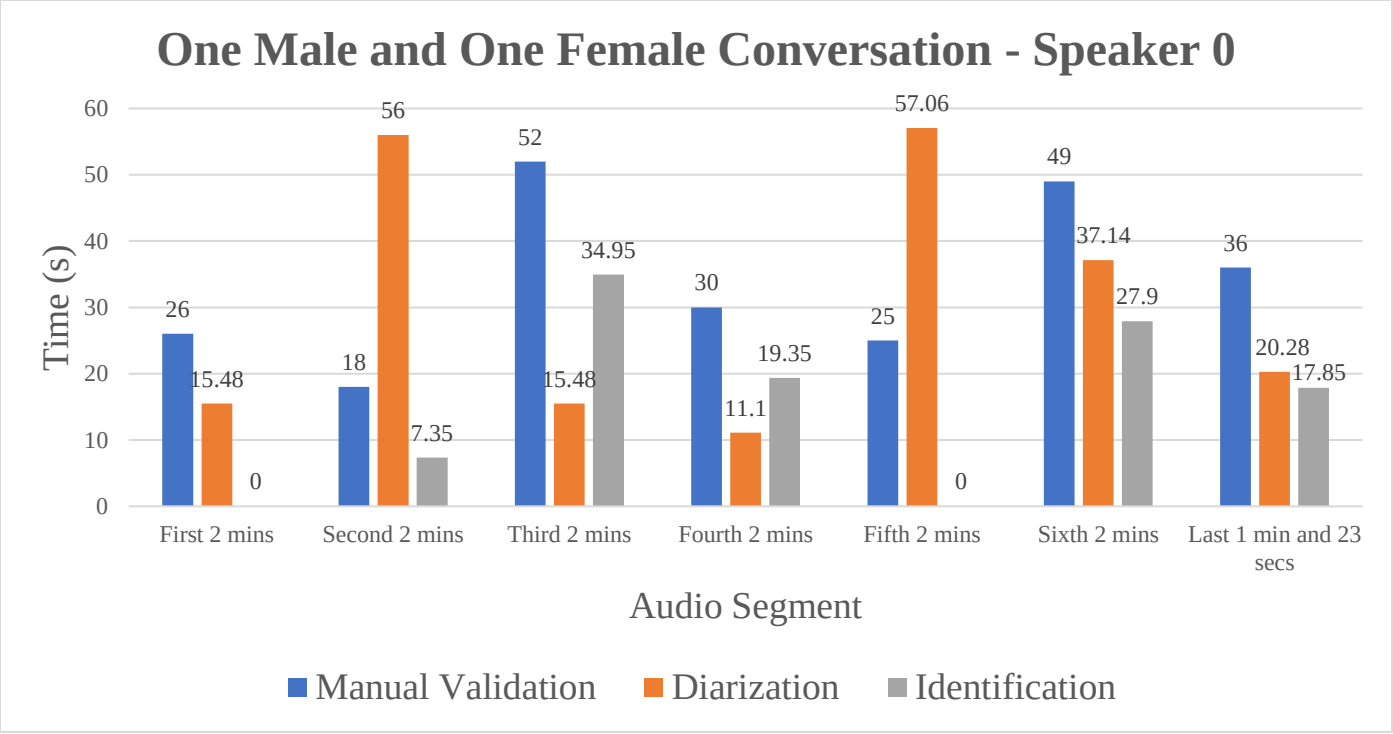

Figure 6.3.8.1 One Male and One Female Conversation – Speaker 0 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	94	53.06	100.95
Second 2 minutes	102	23.76	95.1
Third 2 minutes	68	70.34	52.2
Fourth 2 minutes	90	49.46	68.4
Fifth 2 minutes	95	38.54	95.4
Sixth 2 minutes	71	44	65.85
Last 1 minute and 23 seconds	47	40.28	17.85
Total 13 minutes and 23 seconds	567	319.44	495.75

Table 6.3.8.2 One Male and One Female Conversation – Speaker 1

The total time spoken calculation through Manual Validation for Speaker 1 in this audio file yields a result of 567 seconds compared to values of 319.44 seconds and 495.75 seconds for Diarization and Identification, respectively. Diarization results are lower than Manual Validation results because this conversation has a lot of silences and pauses during the back-and-forth exchange which are trimmed out by the VAD in the Diarization algorithm. Identification results are close to the Manual Validation results for this audio file. The largest discrepancy for Identification occurs in the last 1 minute and 23 seconds audio segment. Identification performed better for Speaker 1, the target speaker, than Speaker 0 in this audio file.

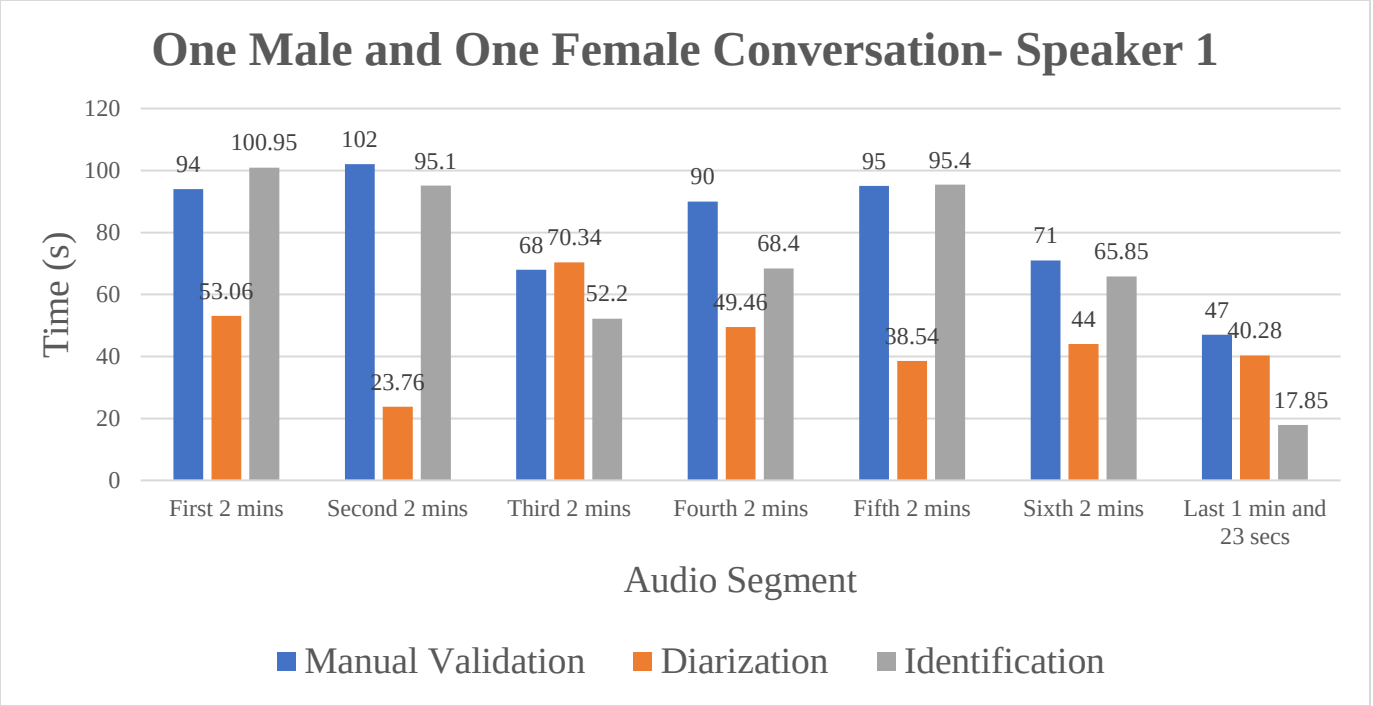

Figure 6.3.8.2 One Male and One Female Conversation – Speaker 1 chart

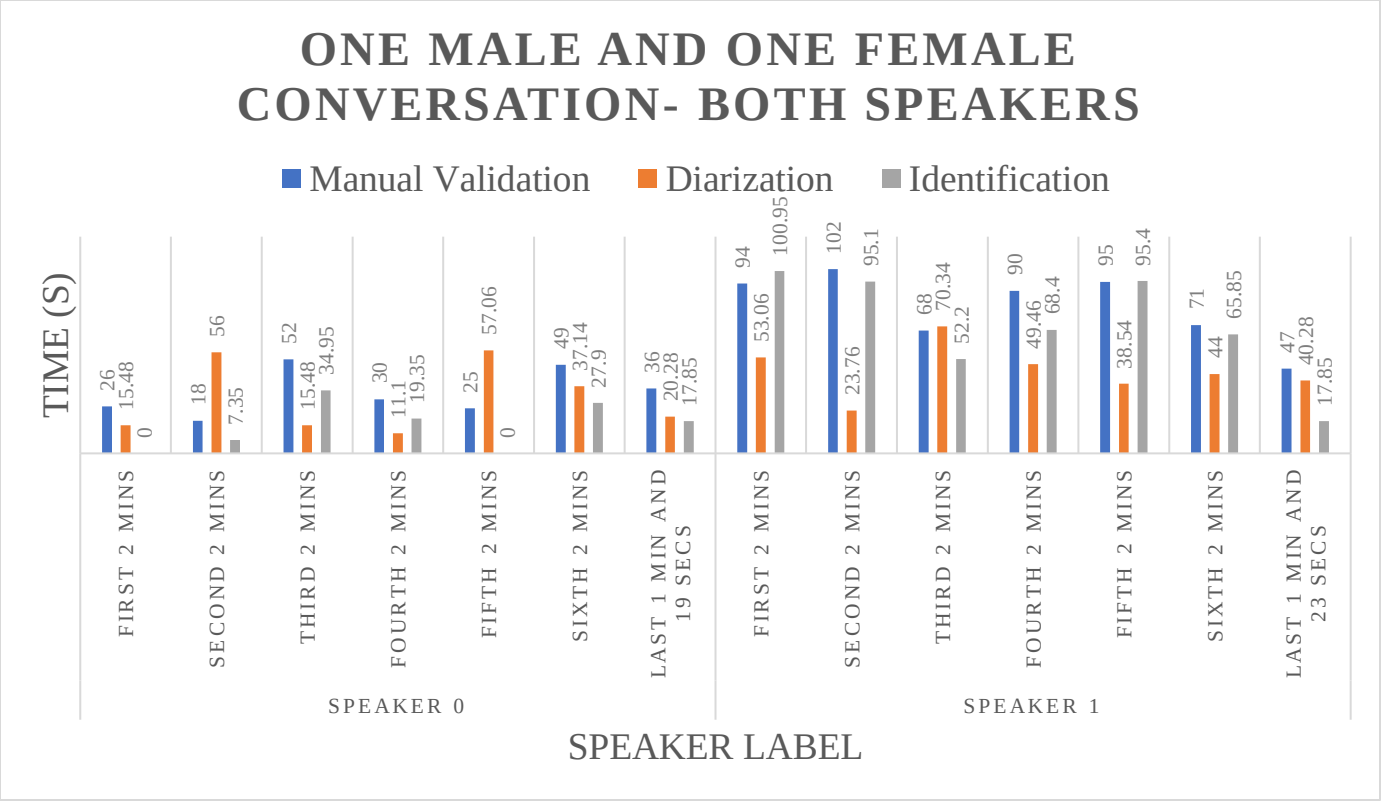

Figure 6.3.8.3 One Male and One Female Conversation– Both Speakers chart

The audio file with one male and one female speaker in a conversational setting was collected with our voice recorder Android application. The results are significantly better for Speaker 1 than Speaker 0 because Speaker 1 was our target speaker, and the voice recorder application was installed on Speaker 1's phone. There were a lot of silences and pauses throughout the conversation which are trimmed by the VAD in the Speaker Diarization algorithm. Hence, the total time spoken values for Diarization are significantly lower compared to both the Identification and Manual Validation methods. Identification results were much closer to manual validation results for our target speaker which implies good performance by the Speaker Identification method.

6.3.9 Dataset 9: Two Male Speakers Conversation

Next, we wanted to validate the audio quality of the data collected from our voice recorder Android application. An experiment was conducted where two male speakers with similar voice characteristics engaged in back-and-forth conversation while simultaneously recording with our voice recorder Android application as well as a voice memo recorder iOS native application on the latest iPhone. Both the iPhone and Android device were placed adjacent to each other at an equidistant position from both the speakers. Our voice recorder Android application produced an audio file of type mp4 whereas the voice memo iOS app produced an audio file of type m4a. Both these audio files were converted to wav format using Audacity so they can be processed with the Speaker Diarization algorithm and sent to the Speaker Identification API. The audio quality was slightly better for the audio file recorded on the iPhone compared to the Android device, but the difference was so minimal that it can be negligible. The results from the 6-minute-long audio file collected through our voice recorder Android application are presented in tables 6.3.9.1 – 6.3.9.2 and figures 6.3.9.1 – 6.3.9.3.

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	51	49.34	82.65
Second 2 minutes	52	50.3	72.3
Third 2 minutes	51	48.24	57.75
Total 6 minutes	154	147.88	212.7

Table 6.3.9.1 Two Male Speakers Conversation – Speaker 0

The total time spoken calculated through Manual Validation for Speaker 0 in this audio file yields a value of 154 seconds compared to 147.88 seconds and 212.7 second for Diarization and Identification, respectively. The Diarization algorithm performed very well with a difference of less than seven seconds when compared to the results from Manual Validation. Identification performed poorly compared to Diarization for this audio file. The total time spoken for Identification was very high because Speaker 0 was identified with a high similarity score in most of the 15-second-long segments that were sent to the Speaker Identification API.

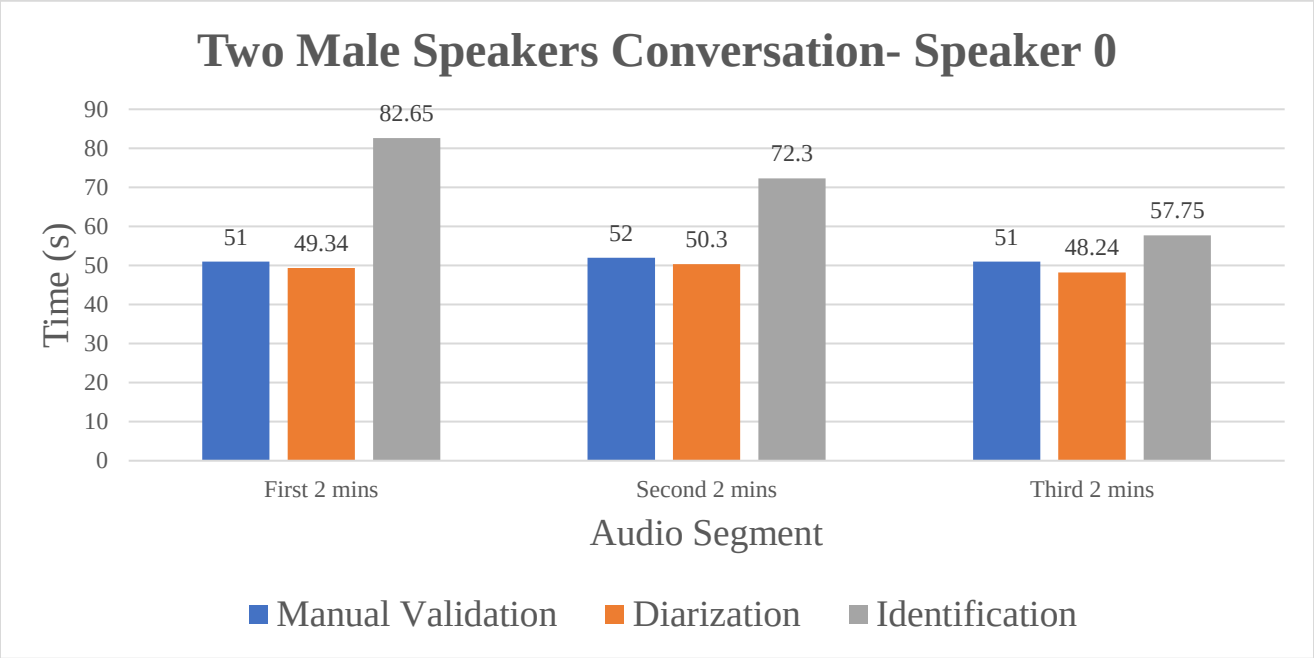

Figure 6.3.9.1 Two Male Speakers Conversation – Speaker 0 chart

Audio Segment	Manual Validation	Speaker Diarization	Speaker Identification
First 2 minutes	58	54.84	77.7
Second 2 minutes	62	57.54	88.2
Third 2 minutes	62	61.82	61.95
Total 6 minutes	182	174.2	227.85

Table 6.3.9.2 Two Male Speakers Conversation – Speaker 1

The total time spoken calculated through Manual Validation for Speaker 1 in this audio file yields a value of 182 seconds compared to values of 174.2 seconds and 227.85 seconds for Diarization and Identification, respectively. Both Diarization and Identification performed well for Speaker 1 with remarkably low error rates which are discussed in further detail in section 6.4 Error Rate.

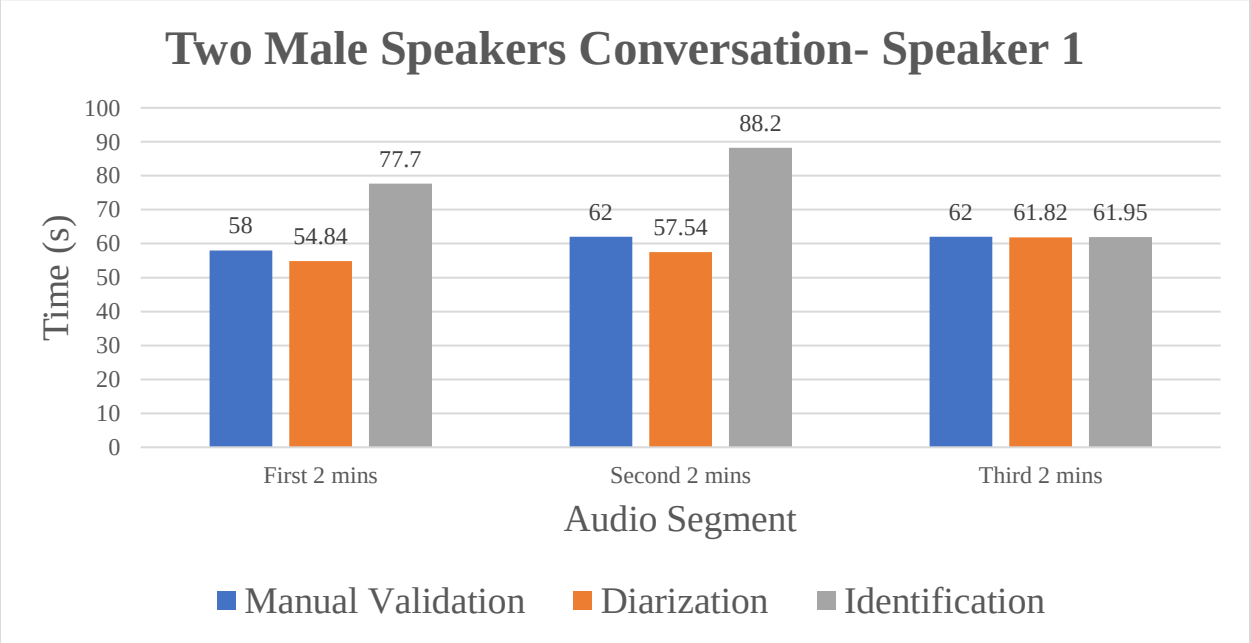

Figure 6.3.9.2 Two Male Speakers Conversation – Speaker 1 chart

The total time spoken splits for both the speakers in the audio file with two male speakers with similar voice characteristics is shown Figure 6.3.9.3. Diarization performed better than Identification for both speakers in this audio file.

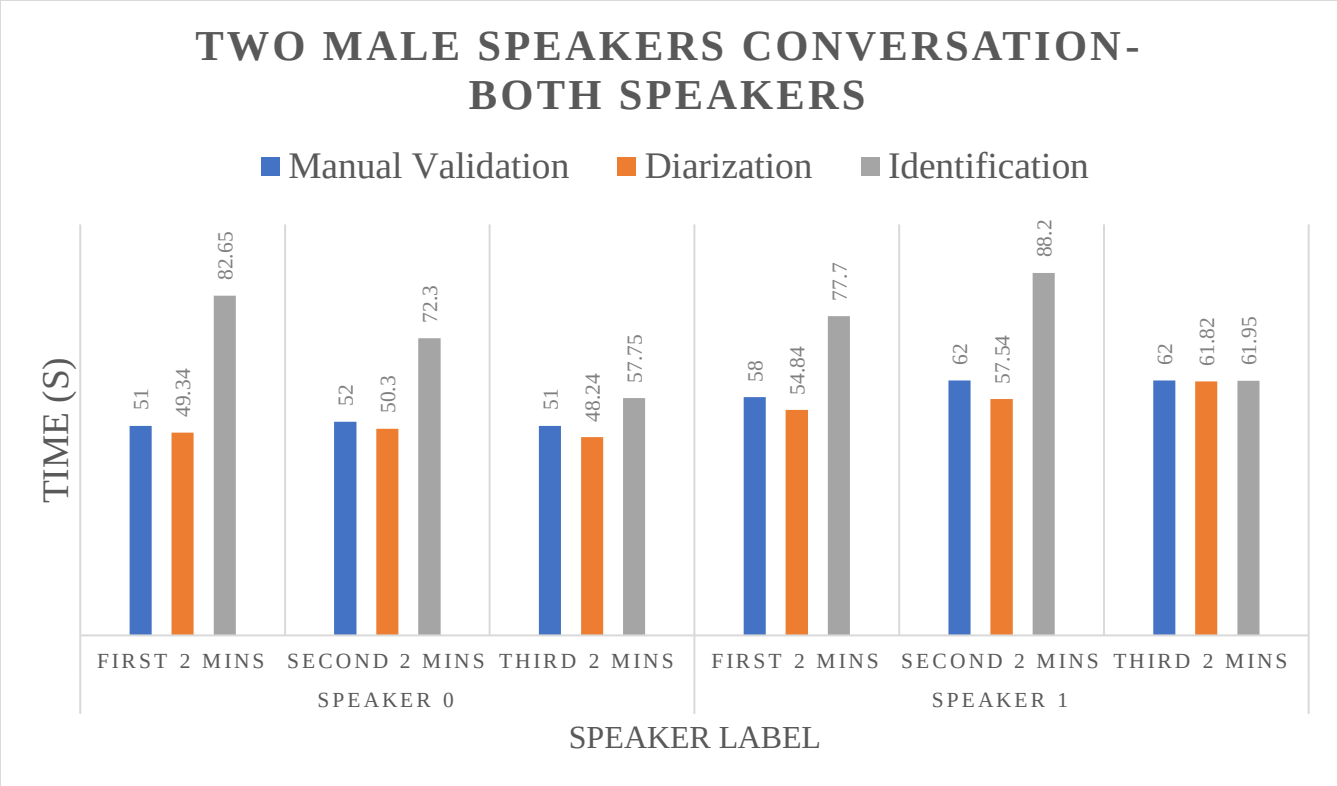

Figure 6.3.9.3 Two Male Speakers Conversation – Both Speakers chart

6.4 Error Rate

In this section, the error rate for Speaker Diarization and Speaker Identification is discussed for audio files from both the Synthetic Diarization Corpus (SDC) and audio files collected by our research team. The error rate per speaker is calculated for each of the audio segments (i.e., first 2 minutes, second 2 minutes, etc). Next, the weighted average of each audio segment relative to the total length of the audio file is calculated for each speaker to determine the average error over the entire audio file. The formula for weighted average calculation is:

$$Weighted\ Average\ = \sum_{first\ audio\ segment}^{last\ audio\ segment} \frac{audio\ segment\ length}{total\ audio\ file\ length} \times error$$

This formula indicates that we begin from the first 2-minute segment and take the summation of all the segments till the last segment in an audio file. The fraction indicates the weight for that segment which is multiplied by the error for a speaker in that audio segment. Essentially, we are calculating the summation off all the errors for a speaker throughout all the segments in the audio file while accounting for the weight of each segment. For instance, consider an audio file with a length of 6 minutes and 14 seconds which converted to seconds will be 374 seconds. The weight of the first two minutes segment would be 120 seconds divided by 374 seconds. The weights change based on the length of the audio segment that the error represents. For example, this audio file of length 6 minutes and 14 seconds can be split into 3 segments of two 2-minute-long segments and one 2-minutes and 14-second-long segment. The weight of the first two segments would be 0.32 since they are equal in length but the weight of the third segment would be 0.36

since it is 14 seconds longer than the other two segments. This allows for an accurate representation of error relative to the duration of which it was calculated for.

Total error was also calculated by taking the difference between the total time spoken of both Diarization and Identification from Manual Validation in the entire audio file for each speaker. Tables 6.4.1 – 6.4.9 show the total error and the weighted average error rate per speaker for each of the audio files discussed in section 6.3 Results.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	8.71% | 15.16%	8.92% | 11.53%
1	4.70% | 9.27%	3.30% | 15.72%
2	10.57% | 11.76%	12.02% | 22.77%

Table 6.4.1 SDC Audio File - Three Speakers Error Rate

Table 6.4.1 shows the error rate for an audio file obtained from Synthetic Diarization Corpus (SDC) dataset that contains three speakers without any overlap. Both the total error and the average is calculated for each speaker for the entire length of the audio file. Speaker 0 had a total error rate of 8.71% for Diarization, 8.92% for Identification, an average error rate of 15.16% for Diarization, and 11.53% for Identification when compared to the Manual Validation results. Speaker 0 had the lowest average error rate for Identification compared to the other two speakers in this audio file. Speaker 1's total error rates of 4.70% for Diarization, 3.30% for Identification, average error rates of 9.27% for Diarization, and 15.72% for Identification are remarkably low. Speaker 1's average Diarization error rate is significant because it is below the 10% threshold and the lowest average Diarization error rate amongst the other speakers within this audio file.

Even though Speaker 2's average Identification error rate of 22.77% is the highest amongst the three speakers in the audio file, Speaker 2's average Diarization error is the second lowest in this audio file with a 11.76% error rate. The total error rates follow a general trend of being lower than the average error rates. Overall, both the Diarization and Identification methods performed well for all three speakers in this audio file.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	4.38% | 5.56%	1.15% | 15.06%
1	9.36% | 10.01%	6.74% | 38.30%
2	4.98% | 5.23%	3.03% | 25.72%

Table 6.4.2 SDC Audio File - Three Speakers with Overlap Error Rate

Table 6.4.2 shows the error rate for an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset that contains three speakers with overlapped speech. Speaker 0 had a total Diarization error rate of 4.38%, total Identification error rate of 1.15%, an average Diarization error rate of 5.56% and an average Identification error rate of 15.06% for the entire duration of the audio file. Speaker 0's average Diarization error rate is the second lowest amongst the three speakers and only falls short of being the lowest average Diarization error rate for this audio file by only 0.33%. Speaker 0's total and average Identification error rates were the lowest amongst all three speakers in this audio file. Speaker 1 had the highest total and average error rate amongst the three speakers in this audio file for both Diarization (9.36%, 10.01%) and Identification (6.74%, 38.30%). Even though the Identification error rate was a bit higher than

expected, the Diarization error rate was relatively low. Speaker 2 had the lowest average Diarization error rate and the second lowest Identification error rate amongst the three speakers in this audio file. The average Diarization error rate of 5.23% for Speaker 2 indicates very good performance by the Diarization algorithm.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	6.37% | 14.05%	15.3% | 20.35%
1	16.98% | 20.94%	34.50% | 44.71%

Table 6.4.3 SDC Audio File - Two Speakers Error Rate

Table 6.4.3 shows the error rate for an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset that contains two speakers without any overlap. Speaker 0's Diarization and Identification error rate are both lower than Speaker 1's error rates with values of 6.37%, 14.05% and 15.3%, 20.35% compared to 16.98%, 20.94% and 34.50% 44.71% respectively. Identification error rates follow a general pattern of being greater than Diarization error rates. Total error rates are generally lower than the average error rates.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	5.14% | 7.09%	2.39% | 14.39%
1	3.60% | 7.49%	33.10% | 35.70%

Table 6.4.4 SDC Audio File - Two Speakers with Overlap Error Rate

Table 6.4.4 shows the error rate for an audio file obtained from the Synthetic Diarization Corpus (SDC) dataset that contains two speakers with overlapped speech portions. Speaker 0 has a Diarization error rates of 5.14%, 7.09% and Identification error rates of 2.39%, 14.39% for the entire audio file. Speaker 1's average Diarization error rate is close to Speaker 0's with a value of 7.49%. However, Speaker 1's average Identification error rate with a value of 35.70% is more than two times greater than Speaker 0's Identification error rate. The total error rates continue to be remarkably low for both Diarization and Identification. Speaker 0 showed great performance for both Diarization and Identification. Speaker 1 also had great performance for Diarization, but Identification error rate is relatively high.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	4.44% | 4.44%	114.56% | 114.56%
1	0% | 0%	32.04% | 32.04%
2	19.55% | 19.55%	24.09% | 24.09%

Table 6.4.5 Three Female Speakers Conversation Error Rate

Table 6.4.5 shows the error rate for an audio file that was collected by our research team during the initial pilot testing phase. This audio file contains three female speakers with similar voice characteristics. Both the total and average error rates are the same because this audio file was not segmented into 2-minute-long segments since this audio file was only a minute long. Diarization performed well for all three speakers especially for Speaker 0 and Speaker 1 where the error

rates are 4.44% and 0% respectively. However, identification did not perform as well for Speaker 0 as it did for Speaker 1. Speaker 2's Identification error rate of 24.09% is the lowest amongst the three speakers in this audio file.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	16.58% | 16.58%	0.75% | 0.75%
1	29.64% | 29.64%	32.5% | 32.5%

Table 6.4.6 Two Female Speakers Conversation Error Rate

Table 6.4.6 shows the error rate for an audio file that was collected by our research team during the initial pilot testing phase. This audio file contains two female speakers with similar voice characteristics. The total and average error rates are the same for this audio file because it was not segmented since it is only roughly 1 minute long. Both Diarization and Identification performed better for Speaker 0 compared to Speaker 1 with Speaker 0 error rates of 16.58% and 0.75% compared or Speaker 1's error rates of 29.64% and 32.5%.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	3.44% | 3.44%	9.21% | 9.21%

Table 6.4.7 One Female Speaker Monologue Error Rate

Table 6.4.7 shows the error rate for an audio file that was collected by our research team during the initial pilot testing phase. The total and average error rates are the same for this audio file because it was not segmented since it is only roughly a minute long. This audio file contains only one female speaker and Diarization and Identification error rates are 3.44% and 9.21% respectively. There was a Diarization error rate because the Diarization algorithm trims out the silences and non-speech portions of the audio, so the total time spoken is less than the value calculated through Manual Validation. Identification error rate below 10% is acceptable because it is an estimate based on the similarity scores in the 15 second segments that a certain speaker is identified. This makes it challenging for the total time spoken calculation from Identification to exactly match the total time spoken calculation from Manual Validation.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	9.94% | 84.78%	54.49% | 60.57%
1	43.66% | 41.26%	12.57% | 11.34%

Table 6.4.8 One Male and One Female Conversation Error Rate

Table 6.4.8 shows the error rate for an audio file that was recorded on our voice recorder Android application. This audio file is 13 minutes in length and contains one male and one female speaker engaging in back-and-forth conversation. The audio quality was poor with a lot of noise and Speaker 0 could not be heard through certain periods of the audio file. Our target is Speaker 1 since the voice recorder Android application was installed on Speaker 1's Android device. Speaker 1's voice was clearer throughout the audio file partially due to the being closer to the recording device. The results support this with both Diarization and Identification error

rates for Speaker 1 being remarkably low. The error rates are greater than the other audio files that were processed because of the poor audio quality paired with the device receiving notifications, vibrating, and moving which can be expected in a realistic scenario.

Speaker Label	Diarization Total | Average Error	Identification Total | Average Error
0	3.97% | 3.98%	38.12% | 38.11%
1	4.29% | 4.31%	25.19% | 25.44%

Table 6.4.9 Two Male Speakers Conversation Error Rate

Table 6.4.9 shows the error rate for an audio file that was collected through our voice recorder Android application. This audio file contains two male speakers with similar voice characteristics engaging in back-and-forth conversation. Diarization achieved the best results for this audio file, and this can be supported with low Diarization error rates 3.98% and 4.31%. The average Identification error rates of 38.11% and 25.44% for Speaker 0 and Speaker 1 respectively are higher than the average Diarization error rates. The values for both total and average error rates are close for this audio file. The identification error are higher because of the speakers being identified with a high similarity score which has an impact on the total time spoken calculation.

6.5 Discussion

Overall, both the Speaker Diarization algorithm and the Speaker Identification system were tested on various audio files which emulate different scenarios that could ultimately affect the performance of the proposed methods. Four different audio files containing either two or three speakers with and without overlap were obtained from the Synthetic Diarization Corpus (SDC) dataset. Five different audio files were also collected by our research team using different audio recorders including the voice recorder Android application that was developed for this research study. The quality of the data collected through our voice recorder Android application was tested through an experiment of collecting the same audio on a different device which has a state-of-the-art voice recorder using the latest hardware available in a mobile device. The performance of Speaker Diarization increases significantly when the number of speakers in the audio file is set a priori through the adjustment of the minimum and maximum number of clusters in the spectral clustering algorithm. Diarization struggles in performance on audio files emulating a realistic conversational exchange because there are a lot of natural pauses and silences in a real conversation which gets trimmed by the VAD in the Diarization algorithm. The Diarization algorithm shows outstanding performance in synthetic audio files like the ones obtained from SDC due to the lack of pauses in the audio signal since the next speaker is ready to speak as soon as the previous one finishes. Speaker Identification performs the best with good quality enrollment audio preferably in the same environment as the recognition audio. Identification excels in performance in realistic audio data especially if there is no overlap in the audio signal. Overall, Identification error rates are much higher than Diarization error rates because the total time spoken calculation for Identification is estimated based on the similarity

score and the number of 15-second-long segments that a speaker was identified throughout each of the 2-minute-long segments in the entire audio file. Manual Validation was done through listening to the audio file and identifying the period in which a certain speaker spoke and calculating a sum of those times to find the total time spoken for each speaker. So naturally, the error rate for Identification was higher than Diarization when compared to Manual Validation. The performance of both Diarization and Identification does not get compromised even when processing audio files with speakers that have very similar voice characteristics. The lowest Diarization error rate was 0.29% and we achieved an error rate of 0.08% for Identification while processing a certain 2-minute-long segment within the audio file containing two male speakers with similar voice characteristics engaging in a conversation. The audio files that were collected by our research team showcased numerous challenges while collecting audio data such as poor audio quality due to noise from the movement of the recording device, vibration due to the recording device receiving notifications and the distance between the microphone in the recording device from the speakers. The quality of the audio has a direct relation to the accuracy of both Diarization and Identification. Therefore, it is essential to maintain good audio quality during the data collection process. The results from section 6.3.8: Dataset 8: One Male and One Female Conversation provide a lot of valuable insights when faced with these challenges surrounding poor audio quality. The target speaker had much better performance for both Diarization and Identification because the voice recorder Android application was installed and was recording on the target speaker's mobile device which was closer in proximity to the target speaker. As a result, the audio quality of the other speaker was very poor which led the other speaker to be barely audible throughout certain segments of the audio file. Therefore, this other speaker was not identified by the Speaker Identification API during certain audio segments.

Identification was impacted more than Diarization in this scenario since the performance of the Diarization algorithm can be tuned based on the quality of the data being processed whereas Identification is a black box since Microsoft does not give access to the algorithms and models. Another key observation is that the total error rates are significantly lower when compared to the weighted average error for the audio files obtained from SDC, but both the total and weighted average error rates were similar for the audio files that were collected by our research team.

7. Conclusion

The challenges faced, an overall summary, and the future work that could be done is discussed in this section.

7.1 Challenges

We faced numerous challenges in the initial development phase in determining the best algorithm and service to use for Identification and Diarization. We explored many cognitive cloud services from providers such as Google Cloud Platform (GCP), Amazon Web Services (AWS) and Microsoft Azure before finding the algorithms and services that best matched our needs.

Speaker Diarization was initially explored through the Google Cloud Speech to Text service, but this service transcribes the audio file during the Diarization process which raised some privacy concerns that arise with decrypting the content of the audio.

We faced various errors and issues while implementing the Speaker Identification method through Microsoft Azure Cognitive Services. The API calls were failing initially due to setting up the service in the US-East region since it was closer geographically. Certain cognitive services by Azure are only offered in the US-West region so we had to compromise on the latency caused due to our physical location even though this had minimal effect on the actual

performance of the system. I used services like Postman to test the API endpoints and examine the returned JSON data to resolve the issues.

Speaker Diarization algorithm performed poorly on certain segments of our dataset and this was due to reasons such as poor audio quality and noisy environment. The specifications of the algorithm had to be adjusted by configuring the minimum number and maximum number of clusters to improve the performance.

7.2 Summary

Both the Speaker Diarization algorithm and Speaker Identification system performed well on all the audio files that we tested. Overall, Diarization has lower error rates than Identification because error rate was calculated in comparison to the Manual Validation results. The Manual Validation process closely resembles the Speaker Diarization algorithm in terms of partitioning the audio files into homogenous segments and assigning a speaker label to each segment. Whereas Identification yields a similarity score for the different 15-second-long segments that were sent to the Speaker Identification API using secure, RESTful API calls. The total time spoken for Identification is calculated by taking a sum of all the similarity scores greater than the minimum acceptable threshold then multiplying that value by 15 to represent each 15 second segment that the speaker was identified in so we estimate the total time spoken for a certain speaker. Therefore, the results for Diarization were much closer to the Manual Validation in comparison to the Identification results. There were certain scenarios where Identification seemed to outperform Diarization, and this was evident in audio files with a lot of pauses

between conversation or a lot of background noise because the Speaker Diarization algorithm utilizes a VAD to trim out all the non-speech portions from the audio file. Therefore, we noticed a pattern of the total time spoken values calculated using the Diarization algorithm being less than both Identification and Manual Validation results. Performance was not affected when the speakers' voice characteristics were similar since the best results observed belonged to the audio file with two male speakers with similar voice characteristics. The lowest recorded error rates were 0.29% and 0.08% for Diarization and Identification, respectively for the third 2-minute-long segment within this audio file with two male speakers with similar voice characteristics engaging in a conversation. The output from the Diarization method requires further processing to determine to whom does a certain speaker label belong to since Diarization only partitions the audio file into homogenous segments associated with each speaker but we still need to determine the identity of the speaker for each speaker label. Diarization is free of cost and we have access to the code since it is our own implementation. Identification could become costly as we scale up and it is a black box since Microsoft does not give access to the code or models behind the Identification API. Identification is essential determining the identity of a speaker given an audio signal. Identification outperforms Diarization in realistic scenarios where there are a lot of pauses or silences which are common in a real conversation. Identification performance improves significantly with good quality enrollment and recognition audio. Overall, both Diarization and Identification are useful since they solve different challenges in the process of determining when our target speaker spoke.

7.3 Future Work

The Speaker Diarization algorithm uses state-of-the-art d-vector embeddings, and spectral clustering algorithm, but the Diarization algorithm performance can be improved as new research studies are conducted and available for use. The Speaker Identification system performance can also improve as Microsoft improves their models and implements newer API versions. We can also conduct more research and implement creative solutions to furthermore automate the process of social interaction measurement and decrease the required manual human intervention in collection and analysis of audio data.

We could also introduce the process of training in our system to cater the performance of the proposed methods to the target user's voice. This may increase the accuracy of identifying our target user in a multi-speaker situation.

Our voice recorder Android application could be improved to reduce the utilization of the system resources on the recording device. This will help improve the audio collection process for our target user due to the minimal invasive nature of data collection.

The voice recorder application can be expanded to wearable technologies to allow more flexibility in the data collection process. We could also address some potential challenges that arise in realistic settings such as poor audio quality due to noise or distance between the microphone in the recording device and speakers. There are numerous other challenges that can be addressed such as the battery life of the recording device while collecting audio over an extended period and the secure storage of this large amount of data while respecting the privacy of the users.

BIBLIOGRAPHY

[1] A. Sehgal and N. Kehtarnavaz, "A Convolutional Neural Network Smartphone App for Real-Time Voice Activity Detection," in IEEE Access, vol. 6, pp. 9017-9026, 2018, doi: 10.1109/ACCESS.2018.2800728.

[2] G. Frewat, C. Baroud, R. Sammour, A. Kassem and M. Hamad, "Android voice recognition application with multi speaker feature," 2016 18th Mediterranean Electrotechnical Conference (MELECON), 2016, pp. 1-5, doi: 10.1109/MELCON.2016.7495395.

[3] A. Awais, S. Kun, Y. Yu, S. Hayat, A. Ahmed and T. Tu, "Speaker recognition using mel frequency cepstral coefficient and locality sensitive hashing," 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), 2018, pp. 271-276, doi: 10.1109/ICAIBD.2018.8396208.

[4] Q. Wang, C. Downey, L. Wan, P. A. Mansfield and I. L. Moreno, "Speaker Diarization with LSTM," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5239-5243, doi: 10.1109/ICASSP.2018.8462628.

[5] A. Zhang, Q. Wang, Z. Zhu, J. Paisley and C. Wang, "Fully Supervised Speaker Diarization," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 6301-6305, doi: 10.1109/ICASSP.2019.8683892.

[6] H. H. Mao, S. Li, J. McAuley, and G. W. Cottrell, “Speech Recognition and Multi-Speaker Diarization of Long Conversations,” *Interspeech 2020*, 2020.

[7] B. Lin and X. Zhang, “Speaker Diarization as a Fully Online Learning Problem in Minivox,” Oct. 2020.

[8] R. Saxena, “Who spoke when? Build your own Speaker Diarization module from scratch!,” *Medium*, 24-Jun-2020. [Online]. Available: https://medium.com/saarthi-ai/who-spoke-when-build-your-own-speaker-diarization-module-from-scratch-e7d725ee279.

[9] Kiliu, “Speaker Recognition APIs,” *APIs | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/rest/api/speakerrecognition/.

[10] P. Nair, “The dummy's guide to MFCC,” *Medium*, 27-Jul-2018. [Online]. Available: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd#:~:text=Mel%20scale%20is%20a%20scale,in%20speech%20at%20lower%20frequencies.

[11] “Mel-frequency cepstrum,” *Wikipedia*, 14-Apr-2021. [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

[12] M. Phi, “Illustrated Guide to LSTM's and GRU's: A step by step explanation,” *Medium*, 28-Jun-2020. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[13] “Understanding LSTM Networks,” *Understanding LSTM Networks -- colah's blog*. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[14] “Pricing - Speaker Recognition API: Microsoft Azure,” *Pricing - Speaker Recognition API | Microsoft Azure*. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speaker-recognition/.

[15] Resemble-Ai, “resemble-ai/Resemblyzer,” *GitHub*. [Online]. Available: https://github.com/resemble-ai/Resemblyzer.

[16] Aahill, “Create a Cognitive Services resource in the Azure portal - Azure Cognitive Services,” *Create a Cognitive Services resource in the Azure portal - Azure Cognitive Services | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/cognitive-services-apis-create-account?tabs=multiservice%2Cwindows. .

[17] E. Edwards, M. Brenndoerfer, A. Robinson, N. Sadoughi, G. P. Finley, M. Korenevsky, N. Axtmann, M. Miller, and D. Suendermann-Oeft, “A Free Synthetic Corpus for Speaker Diarization Research,” *Speech and Computer*, pp. 113–122, 2018.

[18] Emrai, “EMRAI/emrai-synthetic-diarization-corpus,” *GitHub*. [Online]. Available: https://github.com/EMRAI/emrai-synthetic-diarization-corpus.

[19] F. Pedregosa, “Scikit-Learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] A. S. Gillis, “What is REST API (RESTful API)?,” *SearchAppArchitecture*, 22-Sep-2020. [Online]. Available: https://searchapparchitecture.techtarget.com/definition/RESTful-API#:~:text=A%20RESTful%20API%20is%20an,deleting%20of%20operations%20concerning%20resources.

[21] H. Aronowitz and Z. Weizhong, “New advances in speaker diarization,” *IBM Research Blog*, 05-Nov-2020. [Online]. Available: https://www.ibm.com/blogs/research/2020/10/new-advances-in-speaker-diarization/.

[22] “What is Speaker Diarization?,” *Rev*, 11-Mar-2021. [Online]. Available: https://www.rev.com/blog/what-is-speaker-diarization.

[23] “Speaker Diarization,” *Speaker Diarization - NVIDIA NeMo 1.0.0rc1 documentation*. [Online]. Available: https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/stable/asr/speaker_diarization/intro.html.

[24] S. Furui, “Speaker Identification,” *Speaker Identification - an overview | ScienceDirect Topics*. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/speaker-identification.

[25] O. Knagg, “Building a Speaker Identification System from Scratch with Deep Learning,” *Medium*, 03-Oct-2018. [Online]. Available: https://medium.com/analytics-vidhya/building-a-speaker-identification-system-from-scratch-with-deep-learning-f4c4aa558a56.

[26] D. Stevenson, “What is Firebase? The complete story, abridged.,” *Medium*, 25-Oct-2018. [Online]. Available: https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0.

[27] M. Suzuki and K. Gakuto, “IBM Research AI Advances Speaker Diarization in Real Use Cases,” *IBM Research Blog*, 13-Jul-2020. [Online]. Available: https://www.ibm.com/blogs/research/2020/07/speaker-diarization-in-real-use-cases/.

[28] “Autism Spectrum Disorder,” *National Institute of Mental Health*. [Online]. Available: https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd/.

[29] “Quickstart: Using client libraries,” *Google*. [Online]. Available: https://cloud.google.com/speech-to-text/docs/quickstart-client-libraries.

VITA

Akshay Chavakula was born in Visakhapatnam, Andhra Pradesh, India on May 28th, 1998, to Mr. Chinnababu Chavakula and Dr. Neeraja Naidu Chavakula. He received his Bachelor of Science in Computer Science from University of Missouri-Columbia along with three minors in Mathematics, Business, and Information Technology in 2019. During his undergraduate studies, Akshay completed two internships, conducted undergraduate research, and graduated with Latin Honors as well as Honors Scholar designation.