University of Science and Technology of Hanoi

Bachelor thesis

By

Nguyen Vu Bach BI12-044

Information and Communication Technology

Title:

Speaker Recognition

Supervisor: Dr.Tran Hoang Tung - ICT Lab

Hanoi, 2024

Contents

List of Figures

List of Tables

Acronyms

CD Compact disc.

FFT Fast Fourier transform.

HD High-definition.

MFCC Mel-frequency cepstral coefficients.

ML Machine learning.

OvO One-vs-one.

OvR One-vs-rest.

PCM Pulse-code modulation.

SVC Support vector classification.

SVM Support vector machine.

Abstract

Unlike speech recognition, which transcribe speech to text, speaker recognition focuses on determining speaker's identity based on their voice. To be more detailed, it is a wide field of study that involves identifying individuals based on their speech. The applications of speaker recognition can be found in many aspect of life such as biometrics, smart phone lock, television broadcasting, etc... This article will specify in building a system speaker recognition system that identifying speaker's voice from a group of known voices using support vector machine algorithm. Moreover, the length of input data to use for training section is altered to examine the change in machine learning classifier's result. Then, system performance and reaction to speaker's voice in noisy environment and unknown voices are be covered in this article. Finally, the solution for recent problems, future implementation for the system are also noted down.

Key words: speaker recognition, speaker identification, support vector machine, machine learning.

1 Introduction

1.1 Overview

Speaker recognition is a growing field and used in a wide variety of applications such as smart devices and a lot of biometric security systems[4]. It focuses on identifying and verifying individuals based on their voice. In general, voice is any sound produced by humans to communicate using vocal fold vibration, which occurs when air is under pressure from lungs. It conveys the speaker's traits, such as ethnicity, age, gender and feeling.

The term speaker recognition could refer to speaker verification, speaker diarization and speaker identification. They are different problems, use cases of speaker recognition[5]. In fact, the speaker verification objective is to determine the input speech that corresponds to claimed identity[6]. For example, using voice to unlock personal smartphones, cars are well known use cases of speaker verification. For speaker diarization, it partitions an audio stream containing human speeches into homogeneous segments based on identity of each speaker[7]. It is extremely useful for providing the speaker's identity in the middle of the meetings and broadcast news. Lastly, speaker identification determines which voice in a group of known voices best matches with the speaker[8]. Most famous example of speaker identification is identifying customers through service calls.

A speaker recognition system usually has two phases which is enrollment and verification. Enrollment phase captures voice samples and extract voice characteristics to form a voice print or model. The second phase is matching which compares the input voice sample to a list of enrolled voices[4]. Then based on the problem, the system can perform verification, diarization or identification.

Moreover, speaker recognition can also be divided into two categories: text-dependent and text-independent. Text-dependent requires the speaker to say a specific phrase during both enrollment and matching[9]. The specific phrase can be a predefined password which is better for a restricted, high security system. Text-independent is on the other hand, freely speak without any constraints. This category is more convenient but require longer enrollment phase to achieve good performance[10].

1.2 Objective

As previously stated, speaker recognition has many applications, each application has a different implementation, enrollment process and data collection process. However, in this project the main objectives of this project are:

- Building a speaker identification system that is able to distinguish human speech in group of known voice and unknown voice individually. Personally, speaker identification has the widest range of use in reality, potentially developed further as a completed product.
- Implementing in text-independent approach. Since security is not the first priority, stricture is unnecessary. Moreover, text-independent allow more flexible dataset which would reduce time to find dataset.
- Aiming for minimum input needed and process time which is 3-5 seconds of speech length and less than 2 second to give its prediction. This aim is near the standard of speaker identification in reality.

1.3 Background knowledge

1.3.1 Sound features

Sound (or Audio) features are description of sound or an audio signal that can basically be fed into statistical or Machine learning (ML) models to build intelligent audio systems. Audio applications that use such features include audio classification, speech recognition, automatic music tagging, audio segmentation and source separation, audio fingerprinting, audio denoising, music information retrieval, and more[11].

Chromagram Chroma features are a powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave[12].

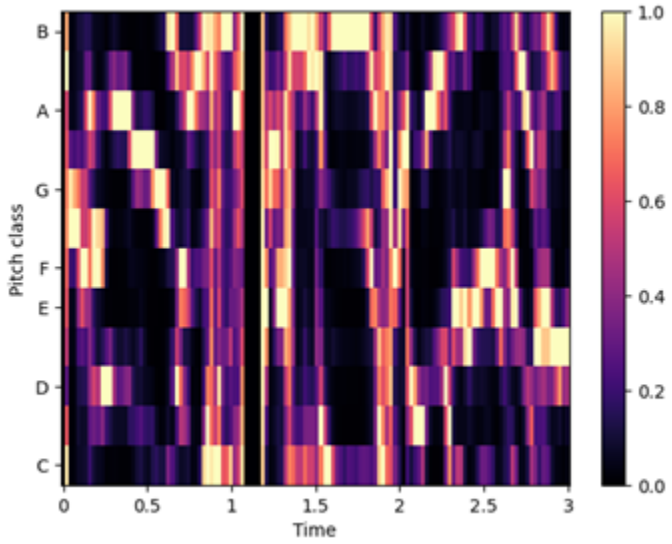


Figure 1: Chromagram of an 3 second audio

Root-mean-square Root mean square is a measurement of how much continuous power an audio signal produces which tells how loud sound is on average over time, taking into account the dynamic range of the signal.[13,1]

$$rms = \sqrt{\frac{2 * (x_1^2 + x_2^2 + ... + x_n^2)}{N}} \tag{1}$$

Spectral centroid The Spectral Centroid provides the center of gravity of the magnitude spectrum. In other words, it gives the frequency band where most of the energy is concentrated. It maps into a very prominent timbral feature called "brightness of sound" (energetic, open, dull). Mathematically, the spectral centroid is the weighted mean of the frequency bins[12,14].

$$centroid = \frac{\sum_{n=0}^{N-1} f(n) * x(n)}{\sum_{n=0}^{N-1} x(n)} \tag{2}$$

Spectral bandwidth The spectral bandwidth or spectral spread is derived from the spectral centroid. It is the spectral range of interest around the centroid, that is, the variance from the spectral centroid. It has a direct correlation with the perceived timbre. The bandwidth is directly proportional to the energy spread across frequency

bands. Mathematically, it is the weighted mean of the distances of frequency bands from the Spectral Centroid[1,14].

$$bandwidth = \sum_{k} S[k,t] * (freq[k,t] - centroid[t]^{p})^{\frac{1}{p}} \tag{3}$$

Roll-off frequency The roll-off frequency denotes the approximate low bass and high treble limits in a frequency response curve, with all frequencies between being those a speaker will play accurately. Anything below or above the rolloff frequency is usually subject to loss of output or inability to perform (blue and white line in the image below).

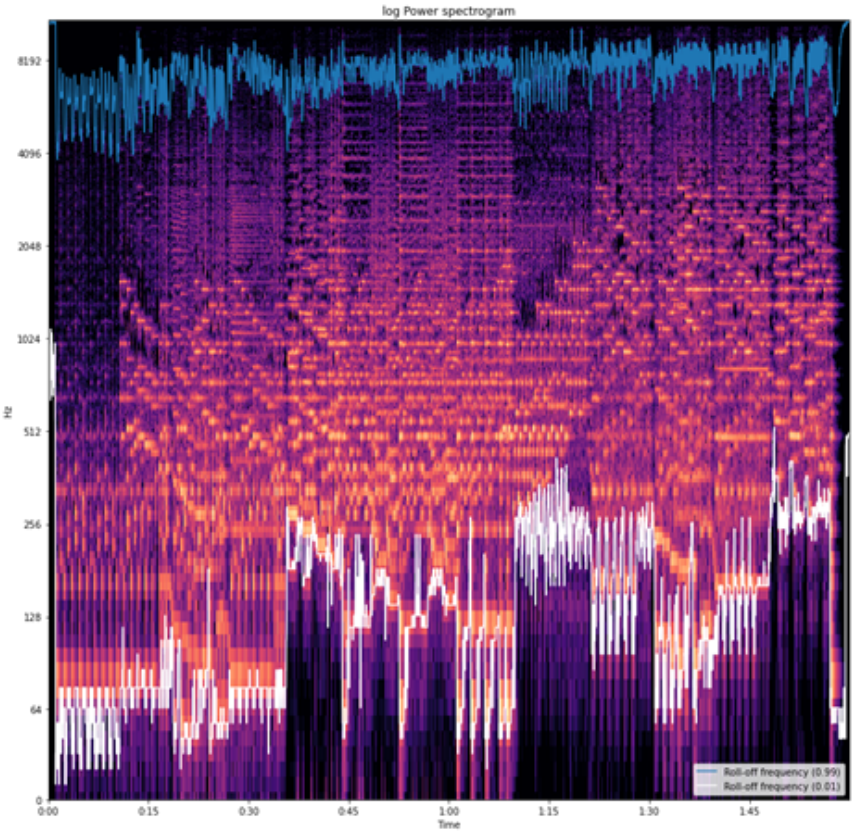


Figure 2: Roll-off frequency of a 1.5 second audio[1]

In each frame, roll-off frequency is defined as the center frequency for a spectrogram bin such that at least 85% of the energy of the spectrum in the frame contained

in the bin[1].

$$rolloff = \frac{1}{T}\sum_{k=1}^{N} E_k \tag{4}$$

Zero-crossing rate Zero-Crossing Rate of an audio frame is the rate of sign-changes of the signal during the frame. In other words, it is the number of times the signal changes value, from positive to negative and vice versa, divided by the length of the frame[1,15].

$$zrc = \frac{1}{1-T}\sum_{t=1}^{T-1} 1_{R<0}(s_t * s_{t-1}) \tag{5}$$

Mel-frequency cepstral coefficients (MFCC) Human hearing is not linear but logarithmic in nature. This implies that our ear acts as a filter. MFCC are based on the known variation of the human ear's critical bandwidth with frequency[16]. Filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in mel-frequency scale. The relationship between frequency in Hz and frequency in Mel scale is given by:

$$m = 2590 \log_{10}(1 + \frac{f}{700}) \tag{6}$$

$$f = 700(e^{\frac{m}{1125}} - 1) \tag{7}$$

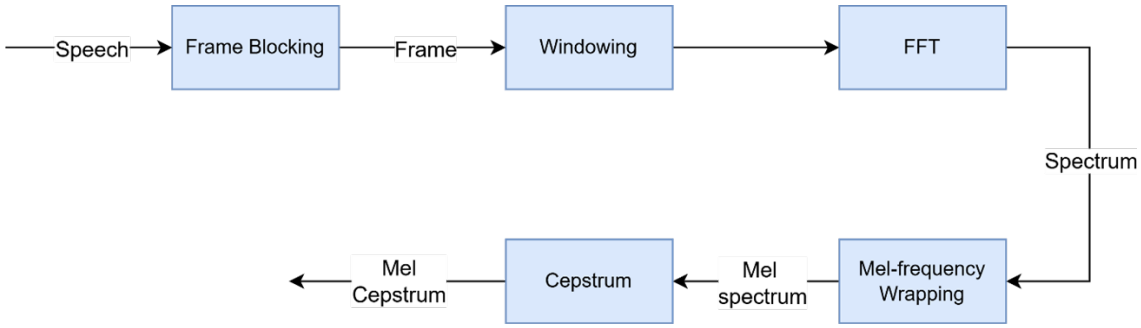


Figure 3: MFCC calculation steps

The speech signal is divided into frames of 25ms with an overlap of 10ms. Each frame is multiplied with Hamming window

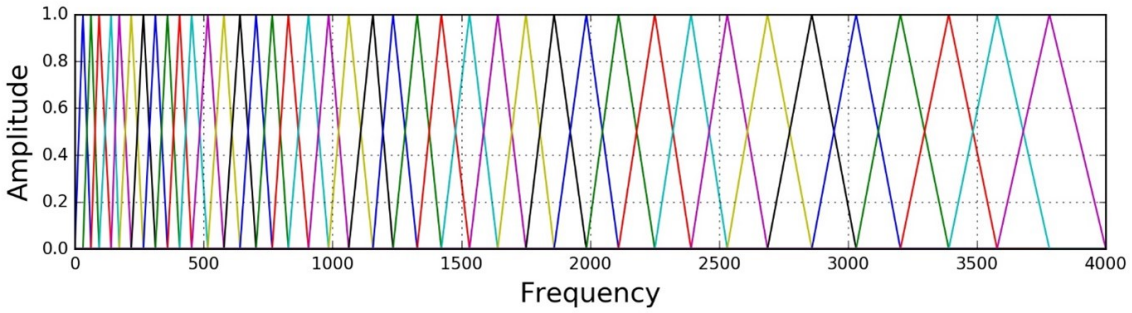


Figure 4: Filter bank [2]

The periodogram of each frame of speech is calculated by first doing an Fast Fourier transform (FFT) of 512 samples on individual frames, then taking the power spectrum as:

$$P(k) = |S(k)|^2 \tag{8}$$

Where P(k) refers to power spectral estimate and S(k) refers to Fourier coefficients for the kth frame of speech and N is the length of analysis window. The last 257 samples of the periodogram are preserved since it is an even function.

The entire frequency range is divided into 'n Mel filter banks, which is also the number of coefficients we want.

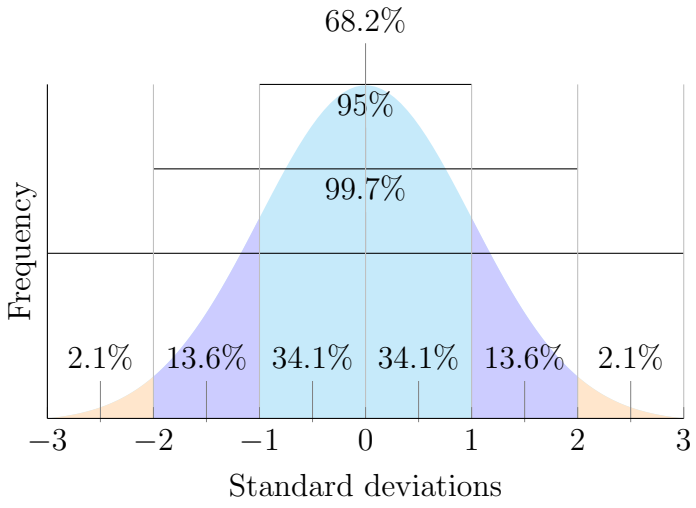
To calculate filter bank energies, we multiply each filter bank with the power spectrum, and add up the coefficients. Once this is performed, we are left with 'n' number that give us an indication of how much energy was in each filter bank. We take the logarithm of these 'n' energies and compute its Discrete Cosine Transform to get final MFCC.

1.3.2 Standardization

Standardizing a vector most often means subtracting a measure of location and dividing it by a measure of scale which also means that features are rescaled around the center and 0 with a standard deviation of 1. This is important when measurements are compared having different units. Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias.

Gaussian distributions are one of the most important distributions in statistics. It is a continuous probability distribution that approximately describes some mass

of objects that concentrate about their mean. The probability density function is bell shaped, peaking at the mean[17].



Standardization assumes that the data has a Gaussian distribution. This does not strictly have to be true, but the technique is more effective if the distribution is Gaussian. Standardization is useful when the data has varying scales, and the algorithm does make assumptions about the data has a Gaussian distribution[18].

$$z = \frac{x - \mu}{\sigma} \tag{9}$$

1.3.3 Support vector machine (SVM)

SVM is supervised learning algorithm that can be used with both regression and classification tasks. It had been used in a wide range of fields because of its amenability to theoretical analysis. In general, the main objective for SVM algorithm is to find the optimal hyperplane in N-dimension space that separates the data point in different classes in the feature space where hyperplane can be written as[19]:

$$w^T x - b = 0 \tag{10}$$

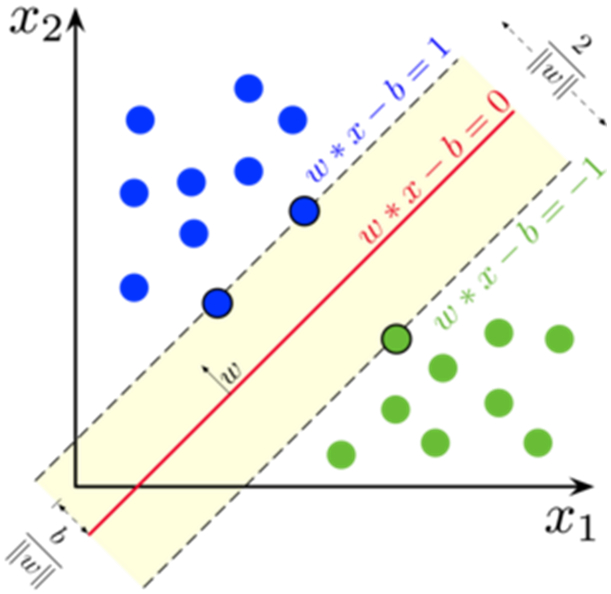


Figure 5: SVM hyperplane[3]

The hyperplane tries to ensure that the margin between the closet point of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the input features is three, then hyperplane becomes a 2D plane.

In classification task, Support vector classification (SVC) performance is more robust and flexible than other liner model such as logistic regression and linear regression. Moreover, support vector classification has two approaches One-vs-one (OvO) and One-vs-rest (OvR).[20] The OvR involves training k classifiers, each distinguishing one class from the rest, while OvO involves training $\frac{k(k-1)}{2}$ classifiers, each distinguishing between pairs of classes.

One-vs-one (OvO) calculation step:

1. For $k$classes, train $\frac{k(k-1)}{2}$ classifiers
2. Each classifiers (i, j) distinguishes between classes i and j
3. For each pair of classes (i, j):
   (a) Extract the subset of data containing only classes i and j

(b) Create a binary label vector $y^{(i,j)}$:

$$y_k^{(i,j)} = \begin{cases} 1 & , \ y_k = i \\ -1 & , \ y_k = j \end{cases} \tag{11}$$

(c) Solve SVM optimization for each classifier:

$$\min_{w^{A,B},b^{A,B},\varepsilon^{(i,j)}} \frac{1}{2} \left\| w^{(i,j)} \right\|^2 + C \sum_{k=1}^{n} \varepsilon_k^{(i,j)} \tag{12}$$

(d) Subject to

$$y_k^{(i,j)}(w^{(i,j)} \cdot x_k + b^{(i,j)}) \geq 1 - \varepsilon_k^{(i,j)} \forall k \tag{13}$$

$$\varepsilon_k^{(i,j)} \geq 0 \forall k \tag{14}$$

4. For a new data point x, each classifier (i, j) make a prediction:

$$\widehat{y}_k^{(i,j)} = \begin{cases} i & , \ f^{(i,j)}(x) > 0 \\ j & , \ f^{(i,j)}(x) \leq 0 \end{cases} \tag{15}$$

5. Use voting scheme where each classifier votes for one its two classes.

6. Assign the class with the most votes:

$$\widehat{y} = arg \max_i \sum_{j \neq i} 1(\widehat{y}^{(i,j)}(x) = i) \tag{16}$$

1.3.4 Machine learning metrics

Accuracy Calculated by divide the proportion of correct predictions among the total predictions. Widely use in machine learning, deep learning to estimate quality of model, system but has limitation with imbalance dataset.

$$Accuracy = \frac{TP + TN}{Total} \tag{17}$$

Precision High precision mean low rate of false alarm. In other word, model predict true, then it likely correct. Extremely important in scenarios like medical diagnose.

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

Recall High recall mean model effectively detect true positive , focus on minimize false positive, ensure mo actual case undetected.

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

F1-Score Combination of precision and recall into one single value. It balances precision and recall, especially when class distribution is uneven[21].

Binary case

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{20}$$

Multi-class case When it comes to multi-class cases, F1-Score should involve all the classes. To do so, a multi-class measure of Precision and Recall is required to be inserted into the harmonic mean. Such metrics may have two different specifications, giving rise to two different metrics: Micro F1-Score and Macro F1-Score

- Micro F1-Score: The idea of Micro-averaging is to consider all the units together, without taking into consideration possible differences between classes. Basically, Micro approach give the Accuracy formula from a different interpretation.

$$MicroAverageF1 = \frac{\sum_{K}^{k=1} TP_k}{GrandTotal} \tag{21}$$

- Macro F1-Score: Macro approach considers all the classes as basic elements of the calculation: each class has the same weight in the average, so that there is no distinction between highly and poorly populated classes. Macro calculate average precision and recall of every class to compute Macro F1-Score.

$$MacroF1Score = 2*(\frac{MacroAveragePrecision * MacroAverageRecall}{MacroAveragePrecision^{-1} + MacroAverageRecall^{-1}}) \tag{22}$$

Confusion matrix A matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance[22].

		Actual	
		Positive	Negative
Predict	Positive	$TruePositive$	$FalsePositive$
	Negative	$FalseNegative$	$TrueNegative$

Table 1: Binary class confusion matrix

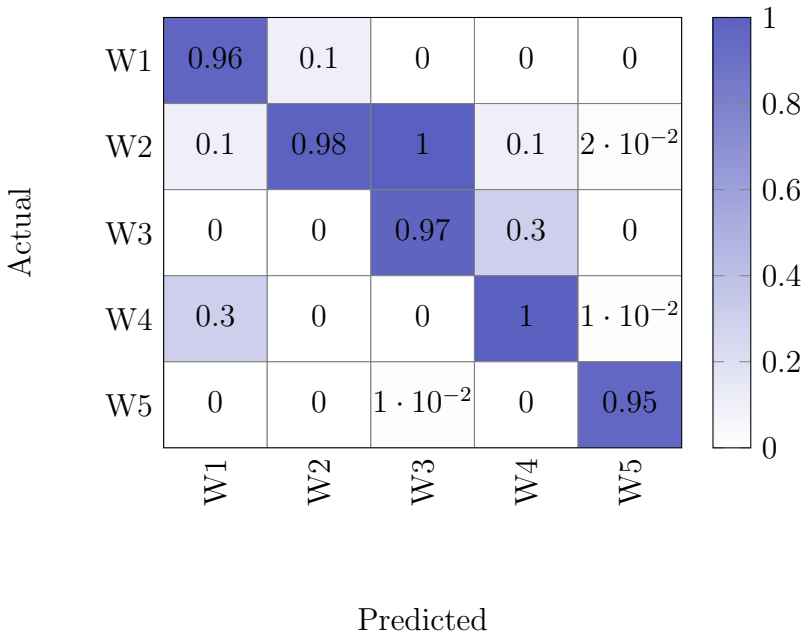


Table 2: Multi-class confusion matrix

1.3.5 Audio data structure

General audio format [23]

- AAC: Apple's alternative to MP3 – stands for 'Advanced Audio Coding'. Lossy and compressed, but sounds generally better. Used for Apple Music streaming
- DSD: The single-bit format used for Super Audio CDs. It comes in 2.8MHz, 5.6MHz and 11.2MHz varieties but, as it's high quality and uncompressed, is impractical for streaming

- MP3: Popular, lossy compressed format ensures small file size but is far from the best sound quality. Convenient for storing music on phones
- WAV: The standard format in which all CDs are encoded. Great sound quality but it's uncompressed, meaning huge file sizes (especially for hi-res files). It has poor metadata support

Wav format[24] WAV files can specify arbitrary bit depth, and this function supports reading any integer Pulse-code modulation (PCM) depth from 1 to 64 bits. Data is returned in the smallest compatible numpy int type, in left-justified format. 8-bit and lower is unsigned, while 9-bit and higher is signed.

Sample rate[25] Sample rate is the number of samples per second that are taken of a waveform to create a discrete digital signal. The higher the sample rate, the more snapshots you capture of the audio signal. The audio sample rate is measured in kilohertz (kHz) and it determines the range of frequencies captured in digital audio.

Bit depth In order to record sound digitally, analogue to digital convertor captures thousands of samples of the analogue signal every second. Each sample is assigned a value which is used to represent the amplitude of that sample. Bit depth defines the number of values that are available in each sample. As a result, bit depth defines the dynamic range of digital audio[26].

Sample rate (kHz)	Description
8	Standard audio for telephony
16	Sufficient for processing human voice
22.5	A popular sample rate
44	Used for compact disc (CD) audio
96	High resolution audio

Table 3: Common sample rate

It's important to know that bit depths come in 2 different forms. The first is 'fixed point', also known as 'integer'. The second is 'floating point'.Unlike fixed point bit depths, floating point bit depths are used only for internal processing[24].

Format	Min	Max
32-bit floating-point	-1	+1
32-bit integer PCM	-2^{31}	$+2^{31}$
24-bit integer PCM	-2^{23}	$+2^{23}$
16-bit integer PCM	-2^{15}	$+2^{15}$
8-bit integer PCM	0	255

Table 4: Some common data type

Channel A digital audio file can contain multiple channels of data. For instance, music that is mixed for headphone listening is saved as a file with two channels - one sent to the left ear, one sent to the right, while surround-sound movie audio is often mixed for 6 channels[27].

2 Material

2.1 Overview

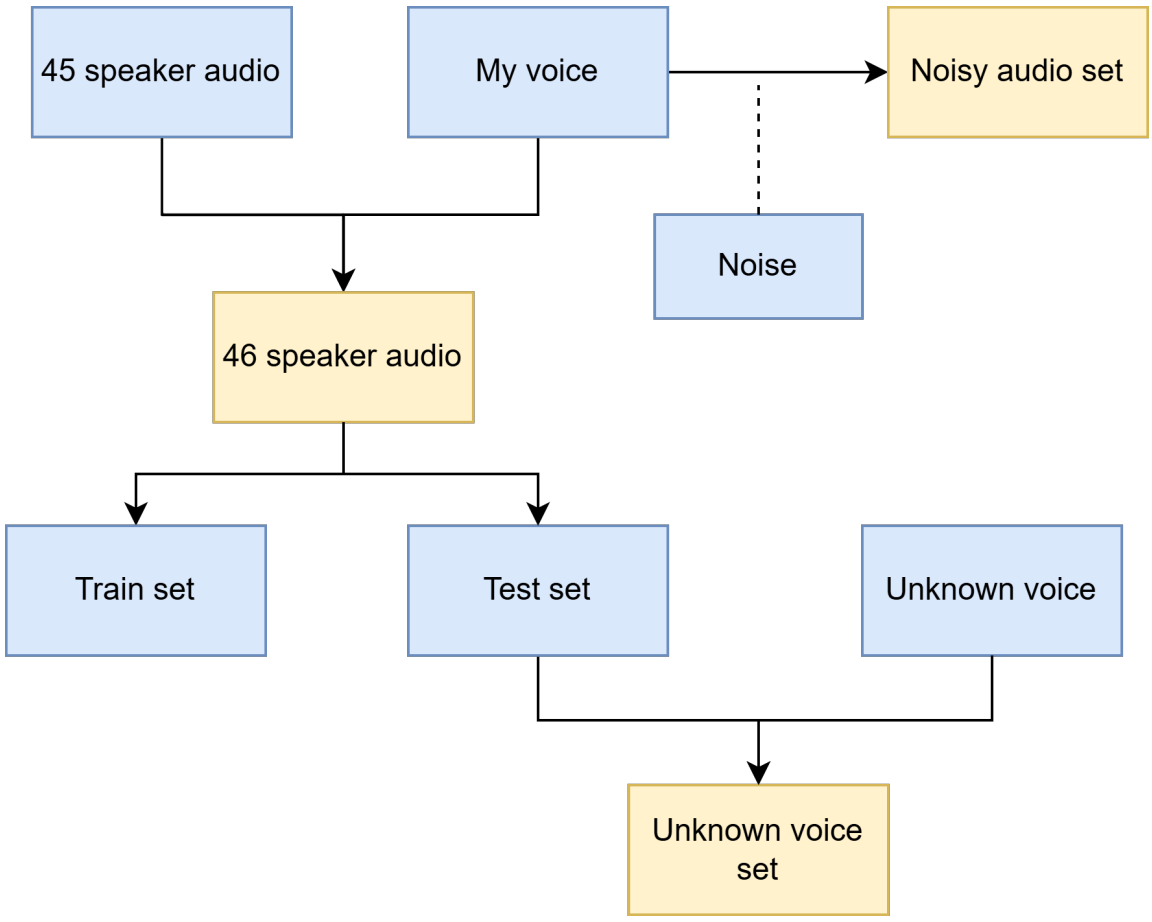


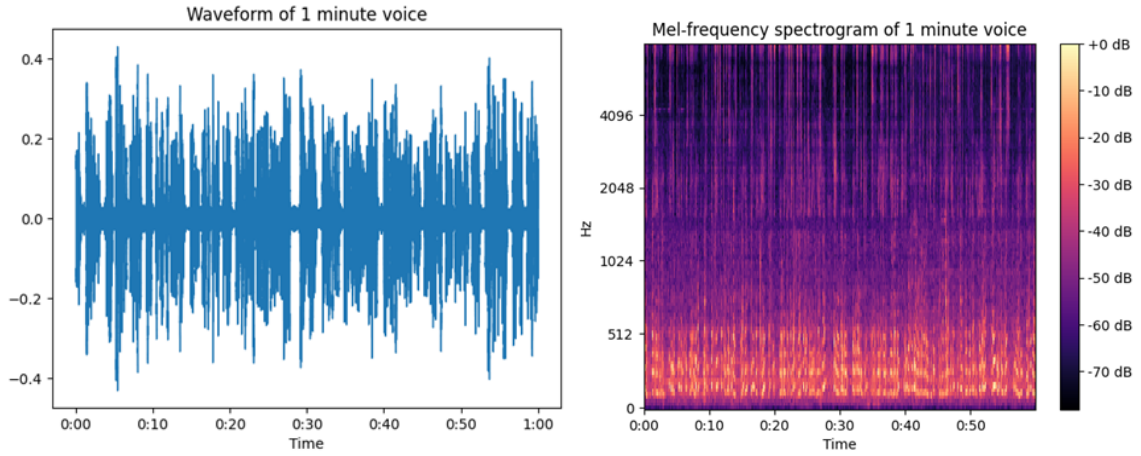
Figure 6: Dataset overview

The figure above show the overview of all the audio sets that used in this project. There are 3 main audio sets which colored in yellow, 2 of them are used just to test the quality of system and last one used for both train and test. Firstly, 46 speaker audio is originally combined using 45 speaker audio and my voice. It then used in Recognition process and split in to train and test set. Secondly, noisy audio set is made using 10 minute of my voice then add 5 different background noises to obtain 5 different version of the voice. The noisy audio set is used to test the ability to recognize speaker's voice in noisy environment. Thirdly, unknown voice set is a

combination of a small part of test set from 46 speaker audio and unknown voice which are untrained voices. The purpose of this set is to test if system can distinguish between known voice and unknown voice.

2.2 46 Speakers audio set

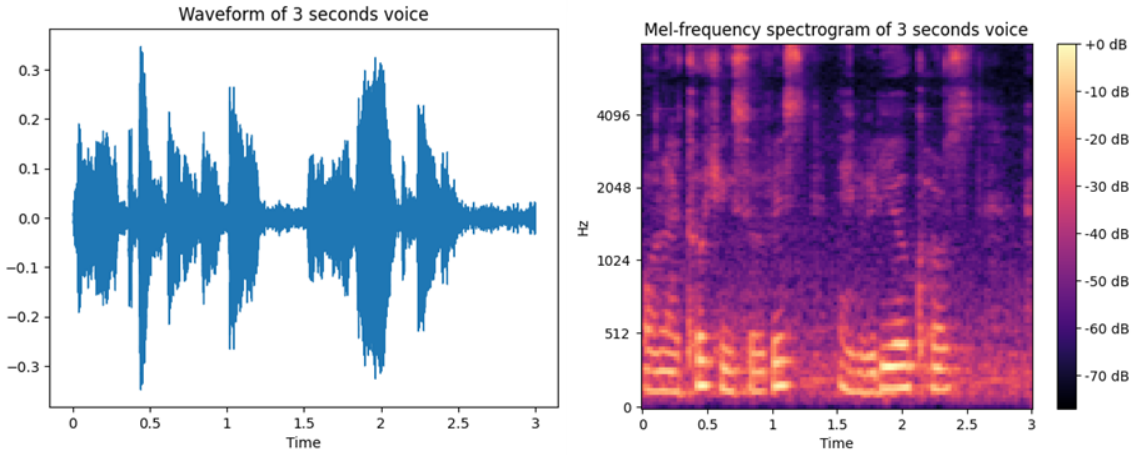
The 45 speakers audio dataset contain 45 difference voices which labeled as speaker 1-45, is originally scraped from YouTube and Librivox with various length from than 30 minutes to an hour for each speakers[28,29,30]. Moreover, my voice was added to the dataset as speaker 46 to form a audio dataset called 46 speaker audio. Further, data converted to wav format, 16kHz-22kHz, mono channel and is split into multiple of 1 min chunks audio. This dataset is suitable for speaker recognition problems.



(a) Waveform of 1 minute audio (b) Mel-spectrogram of 1 minute audio

Figure 7: 1 minute length audio

The average speaking speed is 130-150 words per minute and the average word needed for a sentence is 14-20 words.The desired system is able to recognize a speaker from 3-5 seconds which is 6-14 words and takes 1-2 seconds to give its prediction. That's why the duration of each audio chunk needs to be reduced in order to achieve the final target. Despite being split into 1 minute chunks, the dataset should be splitted into multiple 3 seconds chunks.



(a) Waveform of 3 seconds audio (b) Mel-spectrogram of 3 seconds audio

Figure 8: 3 seconds length audio

2.3 Noisy audio set

As above mentioned, my voice is also added to the dataset as Speaker 46 so that more information about dataset, models and results can be interpreted. My voice was recorded by continuously reading English writing samples which take approximately 50 minutes. For the recording device, Redmi 10 had been used, bit depth is 24 bits and sampling rate of 192kHz[31]. The recording environment was in a quite room where there is no interrupting noise, record voice then save as wav format for further implementation.

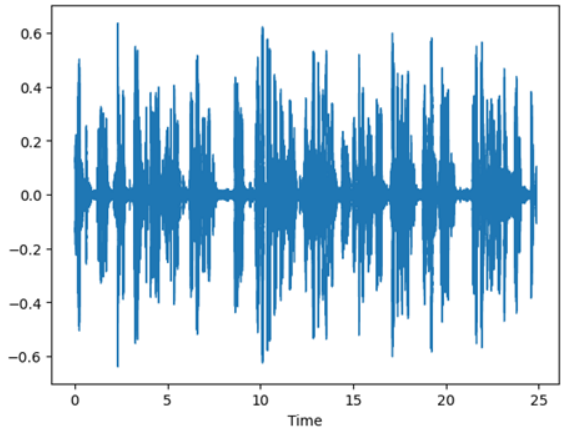
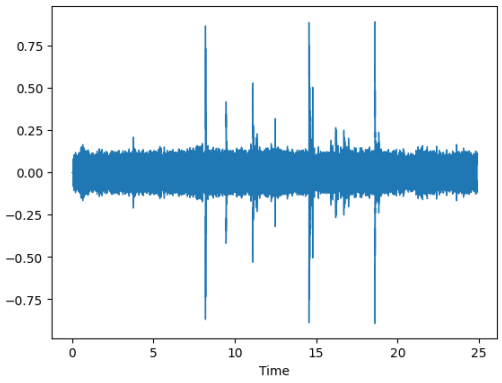
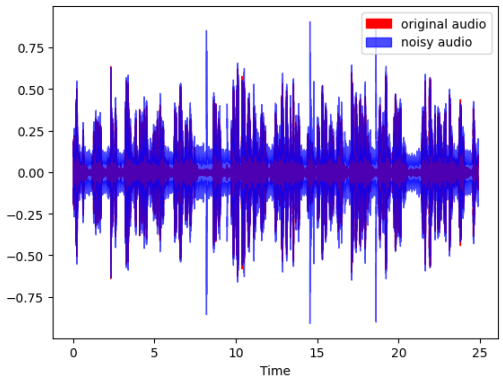


Figure 9: 25 seconds audio

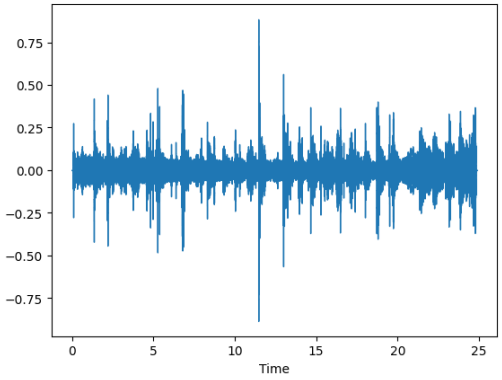
Then, my voice divided in two part. First part has 40 minute length and add to the 45 speaker audio set as speaker46. Second part has about 10 minute length added by noise for testing purpose. The noise are record in 5 different situation: dishwasher, rain, crushing leaves, noisy room and theatre hall[32].



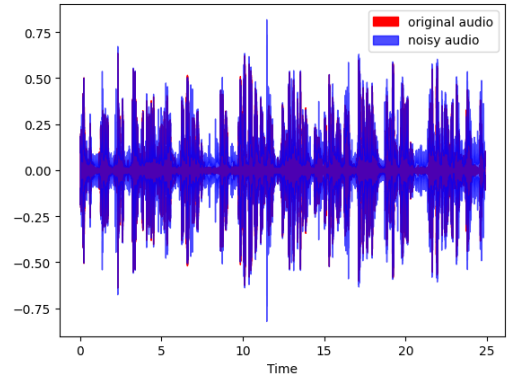
(a) Dishwasher noise



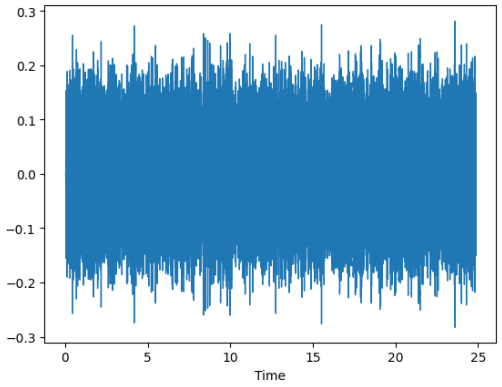
(b) Audio after add dishwasher noise



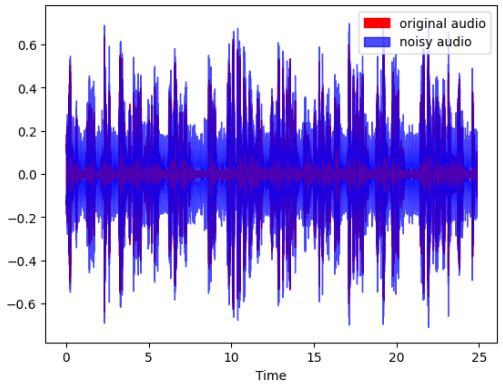
(c) Crushing leaves noise



(d) Audio after add crushing leaves noise

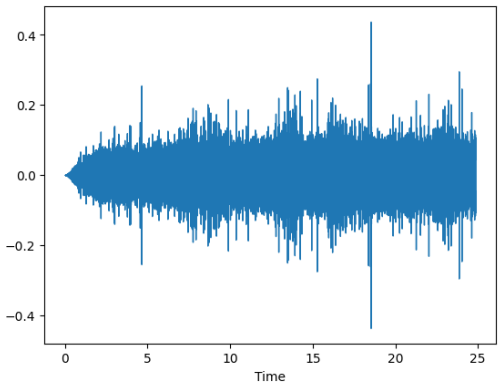


(e) Room noise

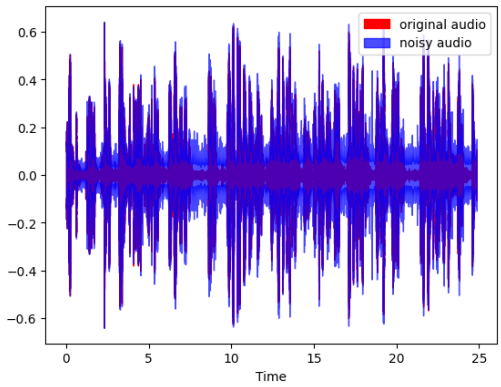


(f) Audio after add room noise

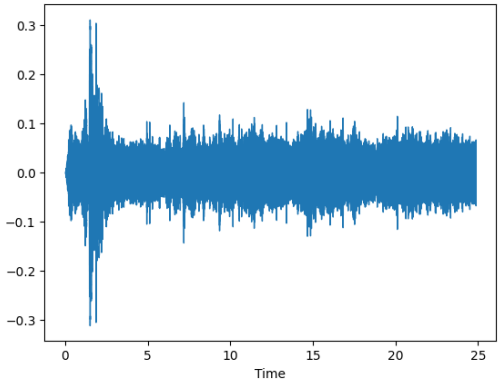
Figure 10: Noise comparison(1)



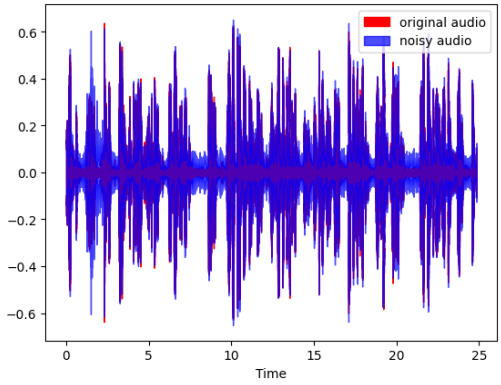
(a) Rain noise



(b) Audio after add rain noise



(c) Theatre hall noise



(d) Audio after add theatre hall noise

Figure 11: Noise comparison(2)

2.4 Unknown set

To test the quality of the speaker recognition model, unknown voice set is also added to the dataset. The unknown voice set must not contain any voice that belongs to any speakers in the original audio set. The set has a total length of over 16 minutes including 6 different voices which are originally from Librivox as well[29].

3 Methodology

3.1 Overall pipeline

As mentioned in overview, generally, a speaker recognition system has 2 phase: Enrollment phase and Recognition phase. Enrollment phase captures voice samples, extract voice characteristics to form a model. Recognition phase directly extract voice characteristics to matching, comparing with enrolled voices to chose the best match voice.

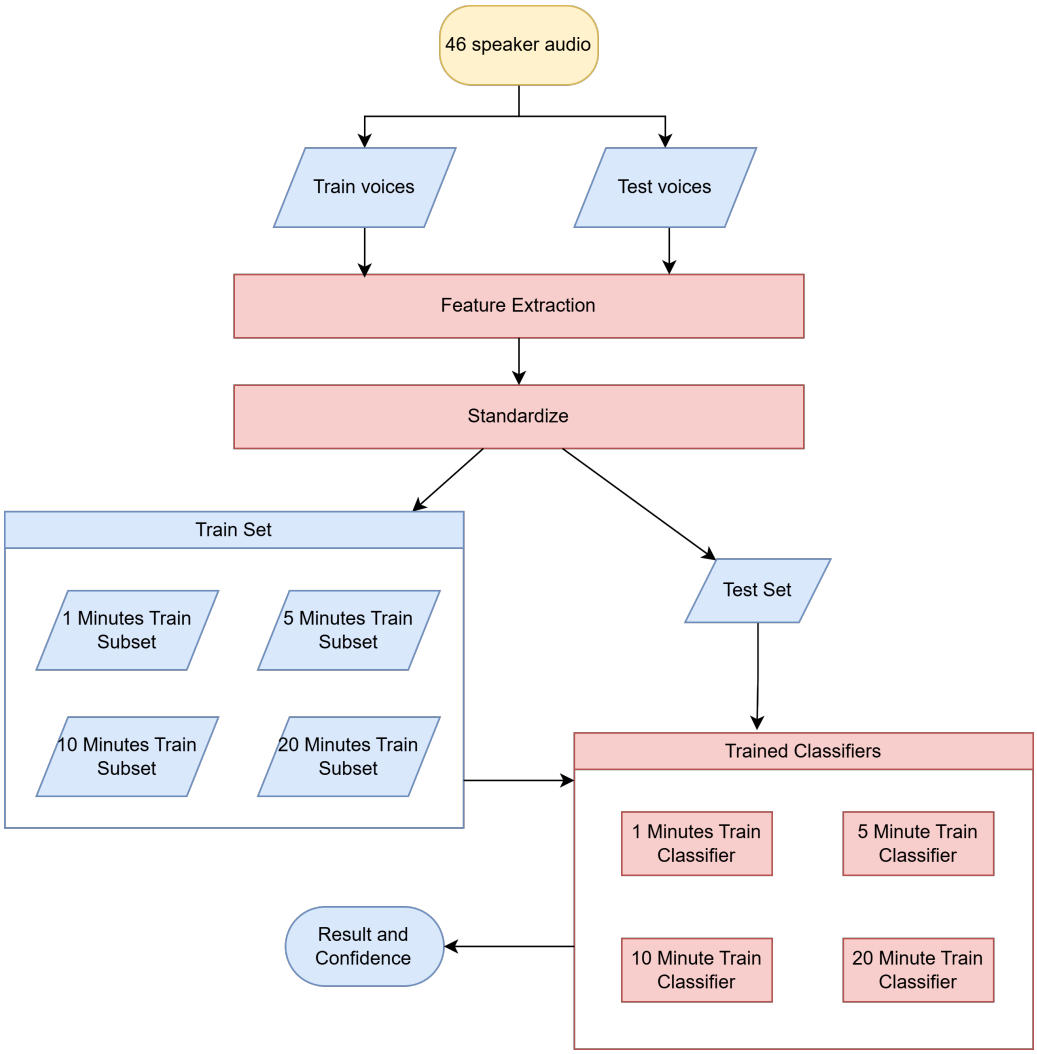


Figure 12: Enroll process

Figure 12 shows the Enrollment process. The main purpose of this process is training model which would later used in Recognition process to distinguish voices. The **46 speaker audio** was chosen for this process and split into **train voices** and **test voices**.

Both voices then extracted into audio features to obtain two dataset that contain information of voice features from train and test voices. Following, both dataset is standardize based on train voices dataset. After that, the train voices set is divided into **4 subsets** with **difference amount of data** while the test voices set remain unchanged. Next, using **SVM** to train each train voices subsets to obtain **4 different classifiers**. The result of classifiers are **best match voice** and a **confidence score** that show how confidence the answer is. Finally, test voices set is used to test the quality of classifiers.

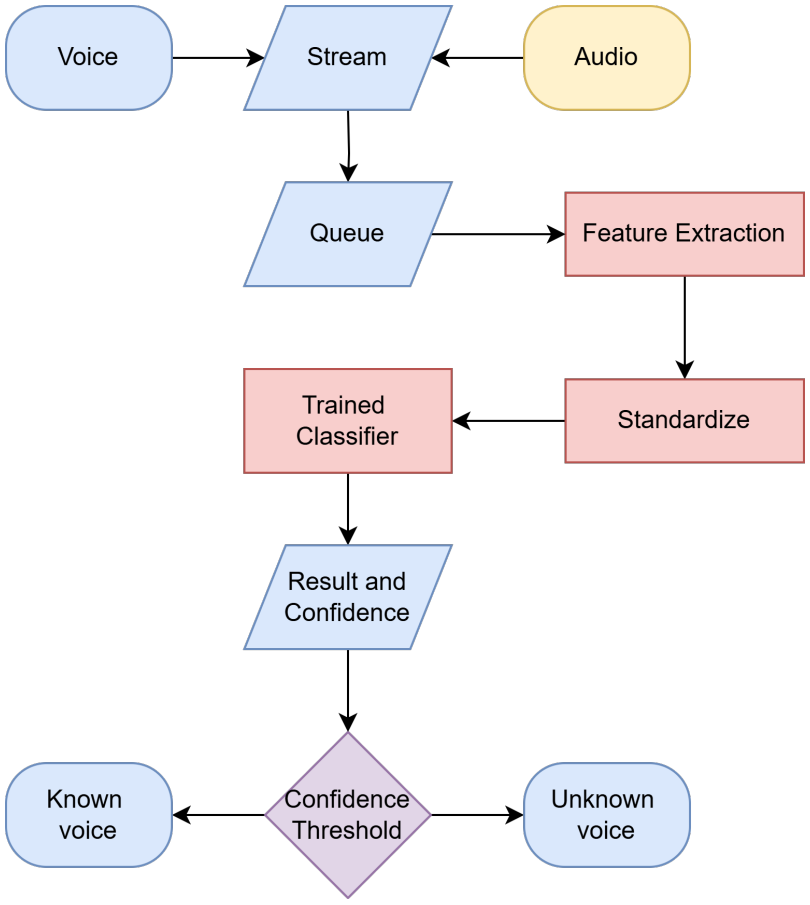


Figure 13: Recognition process

Figure 13 show the Recognition process. The main purpose of this process is to recognize a known voice between a set of known voices; a known voice between a set of unknown voices.

Firstly, stream has two different input. Voice represent real voice captured from device that run this system. Audio represent pre-recorded audio files that read then written into stream. **Audio** is the main input source used for further test. After that, a queue is implemented to capture information from stream. **Every 1 second** of the stream flow, the queue would capture **3 seconds of audio** and use it to run through the rest of process. Following, the 3 seconds of audio would extracted into features, standardize based on train voices set. The result after standardization went to one of the trained classifier that obtain from previous process to obtain best match voice and confidence score. Finally, a **confidence threshold** is implemented to distinguish known voice or unknown voice.

3.2 Enrollment process

3.2.1 Train/Test split

Before train test split, 1 minute audio chunks should be divided into multiple **3 seconds chunks** with no overlap and convert to a sampling rate which is **22.5kHz**. The least audio length of a speaker in the audio dataset is 30 minutes and the most audio length is up to 115 minutes. For that reason, splitting train and test set based on ratio is quite imbalance.

Instead, to ensure the fairness in testing stage and assure that each class has enough data for model training stage, a fixed audio length is taken from each speakers. Based on the shortest audio length of a speaker in this audio dataset which is 30 minutes, **10 minutes** audio of each speakers would be used for **testing purpose** and the remain audio are used for training.

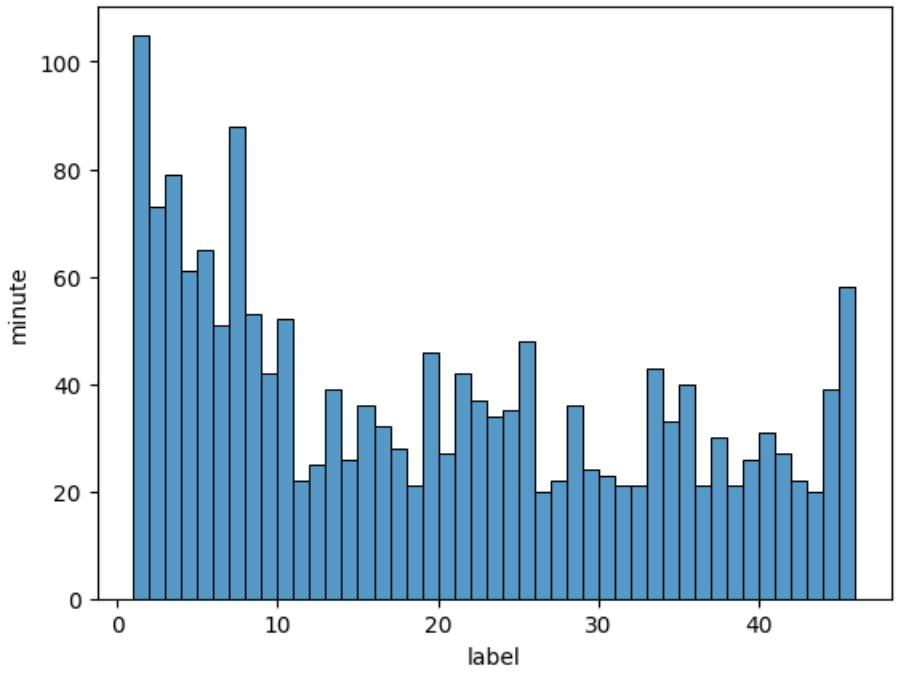


Figure 14: Audio length of each speaker in train set

Figure 14 describes time length of each class in **train set**. The vertical side shows the length of time in minute and the horizontal side shows the classes which also called label.

3.2.2 Feature Extraction

The data is load from audio chunks as 32-bit float. Firstly, float number allow to read digital signals more precisely than integer. Secondly, 32-bit has much wider amplitude which can capture up to 1500dB[33], offers more resolution than other lower bit depth and has lower memory space than 64-bit.

In audio processing, the level of abstraction refers to the different layers at which audio features can be analyzed and understood[11]. Also, audio features can be categorize based on this. In fact, high level feature are easy to recognize by human ear like rhythm, melody, genre. Mid level feature are features that human ear can perceive but difficult distinguish them such as Mel-frequency cepstral coefficients (MFCC), pitch. Finally, low-level are feature are statistical features can only extracted by machine and make sense to machine only like amplitude envelope, energy,

spectral centroid. Since the objective of the project is machine-based the most of audio features should be mid level or lower to **achieve efficiency and accuracy** in machine learning model. Formulas to calculate following features are in 1.3.1 Sound features.

Data	Role	Data type	Description
label	Target	Qualitative	Label of audio
chroma_stft	Feature	Quantitative	Chromagram
rsme	Feature	Quantitative	Root-mean-square
spec_cent	Feature	Quantitative	Spectral centroid
sepc_bw	Feature	Quantitative	Spectral bandwidth
rolloff	Feature	Quantitative	Roll-off frequency
zrc	Feature	Quantitative	Zero-crossing rate
m0	Feature	Quantitative	Mel-frequency cepstral coefficients
m1			
...			
m19			

Table 5: General information about audio features

- **Chromagram**: For the human, voice pitch is an important characteristic because of its ability to guess the gender, age or emotion from that voice. Although there are a several digital signal processing methods to get voice pitch, but chroma feature is chosen for its detail. In fact, chroma is usually used in music analysis. While the pitch is nearly the same in general audio engineering, the equality of sharps and flats does not stand in for music. That is why chroma features is created to distinguish different pitch as detail as possible.
- **Root-mean square**: To achieve the consistency of recording device and recognition system, finding a metric to measure the sound quality is essential. Actually, root-mean square serves as a reliable indicator of performance, extremely helpful in comparing audio components based on average power of audio.

- **Roll-off frequency**: When an audio is unable to fully reproduce, there might be a reason behind this. Eventually, there are a few devices that are unable to fully reproduce the audio due to bad frequency response. They are roll-off frequency which subject to loss due to the inability to reproduce. The roll-off frequency describes how rapid the device attenuates the signal after cut-off.

- **Spectral centroid**: Using the spectral centroid can predict the brightness in an audio file. It is widely used in the measurement of the tone quality of any audio file.

- **Spectral bandwidth**: Bandwidth, a low abtraction level feature is the difference between the upper and lower frequencies in a continuous band of frequencies.

- **Zero-crossing rate**: In short, zero-crossing rate is the number of times a waveform crosses the horizontal time axis. This feature has been primarily used in recognition of percussive vs pitched sounds, monophonic pitch estimation, voice/unvoiced decision for speech signals

- **Mel-frequency cepstral coefficients**: The Mel-frequency cepstral coefficients (MFCC) are nothing but the coefficients that make up the mel-frequency cepstrum. The cepstrum conveys the different values that construct the formants (a characteristic component of the quality of a speech sound) and timbre of a sound.

3.2.3 Standardize

Initially, Support vector classification (SVC) is chosen as main method for recognize speakers. Its decision boundary maximizes the distance to the nearest data points from different classes. Hence, the distance between data points affects the decision boundary SVC chooses. In other words, training an SVC over the scaled and non-scaled data leads to the generation of different models.

The two most widely adopted approaches for feature scaling are normalization and standardization. Normalization maps the values into the [0, 1] interval [34]:

$$z = \frac{x - min(x)}{max(x) - min(x)} \tag{23}$$

Standardization shifts the feature values to have a mean of zero, then maps them into a range such that they have a standard deviation of 1:

$$z = \frac{x - \mu}{\sigma} \tag{24}$$

It centers the data, and it's more flexible to new values that are not yet seen in the dataset. Likewise, it also standardized data to avoid features on the dataset are measured on different scales, to ensure that all features are centered around 0 and have variance in the same order. Therefore, standardization is selected for feature scaling.

Features	Before standardize	After standardize
chroma_stft	$3.05 * 10^{-3}$	1
rsme	$3.05 * 10^{-4}$	1
spec_cent	$3.01 * 10^{5}$	1
sepc_bw	$1.96 * 10^{5}$	1
rolloff	$1.01 * 10^{6}$	1
zrc	$2.22 * 10^{-3}$	1

Table 6: Variance of some features before and after standardize

3.2.4 Model architecture

There are many solutions, methods and algorithms for a classification problem. Firstly, for deep learning approach, most convolution neural network performance and score did not achieve expectation. This due to the amount of data is not big enough for the network to train and reach the desired score. Secondly, for supervised learning approach, there was a few attempt to try different machine learning algorithm like random forest and decision tree and support vector machine. In result, the score of support vector machine when train with train voices set was the best, SVM is chosen as classification algorithm.

Algorithm	Score
Decision tree	0.87
Random forest	0.96
Support vector machine	0.99

Table 7: Score of machine learning algorithm on train voices set

Support vector machine classification calculation step:

1. For 46 classes, train 1035 sub-classifiers
2. Each sub-classifiers (i, j) distinguishes between classes i and j
3. For each pair of classes (i, j):
   (a) Extract the subset of data containing only classes i and j
   (b) Create a binary label vector $y^{(i,j)}$:

$$y_k^{(i,j)} = \begin{cases} 1 & , \ y_k = i \\ -1 & , \ y_k = j \end{cases} \tag{25}$$

   (c) Solve SVM optimization for each classifier:

$$\min_{w^{A,B},b^{A,B},\varepsilon^{(i,j)}} \frac{1}{2} \left\| w^{(i,j)} \right\|^2 + C \sum_{k=1}^{n} \varepsilon_k^{(i,j)} \tag{26}$$

   (d) Subject to

$$y_k^{(i,j)}(w^{(i,j)} \cdot x_k + b^{(i,j)}) \geq 1 - \varepsilon_k^{(i,j)} \forall k \tag{27}$$

$$\varepsilon_k^{(i,j)} \geq 0 \forall k \tag{28}$$

4. For a new data point x, each classifier (i, j) make a prediction:

$$\widehat{y}_k^{(i,j)} = \begin{cases} i & , \ f^{(i,j)}(x) > 0 \\ j & , \ f^{(i,j)}(x) \leq 0 \end{cases} \tag{29}$$

5. Use voting scheme where each classifier votes for one its two classes.

6. Assign the class with the most votes:

$$\widehat{y} = arg\max_{i} \sum_{j \neq i} 1(\widehat{y}^{(i,j)}(x) = i) \tag{30}$$

The result of SVM classification is the class that assigned with the most vote and its proportion which also called confidence score.

3.2.5 Model Training

As Figure 14, the train set of each class consist at least 20 minutes audio length. To examine how good the model according to the amount of data that fit in it, the train set is divided in to 4 smaller subsets. Each subsets take a proportion of train set and they are named by the least trained audio length for each class. The trained classifier of each subset are also named after the minimum trained audio length of each class as well.

Minimum trained audio length (minute)	Maximum trained audio length (minute)	Proportion of train set (%)
1	5.25	5
5	26.25	25
10	52.5	50
20	105	100

Table 8: Train subsets proportion

3.3 Recognition process

3.3.1 Stream

Stream input Stream is a flow of data and the source of data can take directly from speaker device or be written by a pre-recorded audio.

- Real-time voice: Using current speaker device as sound input, the device can capture has up to 48kHz with 24 bit depth which is enough to record a High-definition (HD) audio.

- Audio file: In regard of test the system by current audio dataset, playing it on other device so the current device could capture the audio is not a good choice. Though, audio quality may loss due to many reasons like device output, input or environment. In that case, load the audio file on current device would prevent the quality loss. The audio file is loaded in the same data format, sample rate as enrollment process which is 32-bit float and 22.5kHz. After that, data are written to the stream.

Stream parameters Stream parameters are settings of the stream

- Sample rate: Simply, sample rate is number of frames per second. The higher the sample rate, the more frame captured per seconds. If an audio file with fixed amount of frames is read, increase the sample rate may lead to the reduction of audio length when it is written to stream. As the result, sample rate should be keep the same as enrollment process which is 22.5kHz.

- Buffer size: Using buffer instead of a continuous amount of audio because of processing power. The continuous flow of data being received from speaker device would eat a lot of processor, thus causing potential crashes. Therefore, by storing this into an array. The higher buffer size, the greater latency of audio. Alternately, buffer size too small would increase the process computer has to execute[35]. In this case, buffer size of 1024 samples is chosen for acceptable latency and well process speed.

- Format: Format used in stream is 32-bit float for the same reason as section 3.1.2 Feature extraction.

- Channel: This project does not aim for output audio experience. Because of that, mono channel is enough for playback input audio in real-time.

3.3.2 Queue

Queue implementation is selected for a several reasons. Firstly, the system need a fixed length array to store data from stream, minimum the memory used. Secondly, First-In Last-Out manner from a queue is needed for ability to add data while remove the least recently added data. Lastly, system should able to recognize within 3 seconds of audio, not entire speech then return the result. That's mean the amount of data should correspond to size of queue. Total samples of 3 seconds on 22.5kHz sample rate is calculate by:

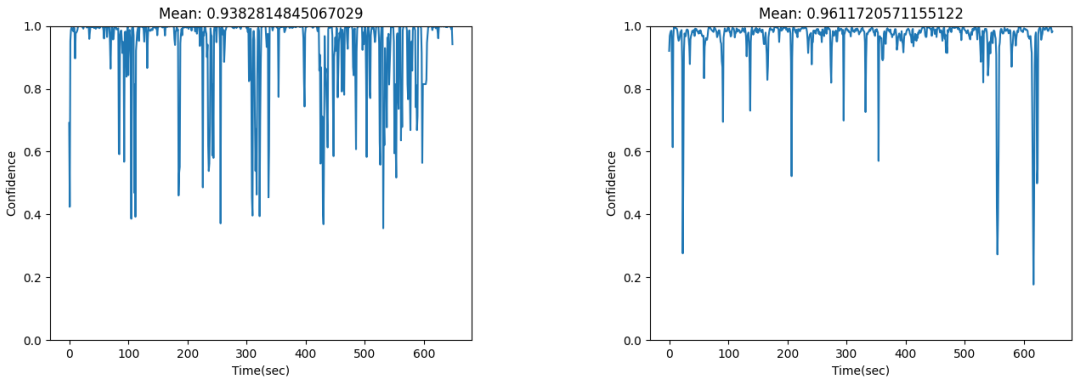
$$TotalSample = SampleRate * Time(sec) = 22500 * 3 = 67500 \tag{31}$$

Then the total sample in 3 seconds is divided by buffer size to get the right length of queue to contain total sample:

$$QueueLength = \frac{TotalSample}{BufferSize} = \frac{67500}{1024} \approx 66 \tag{32}$$

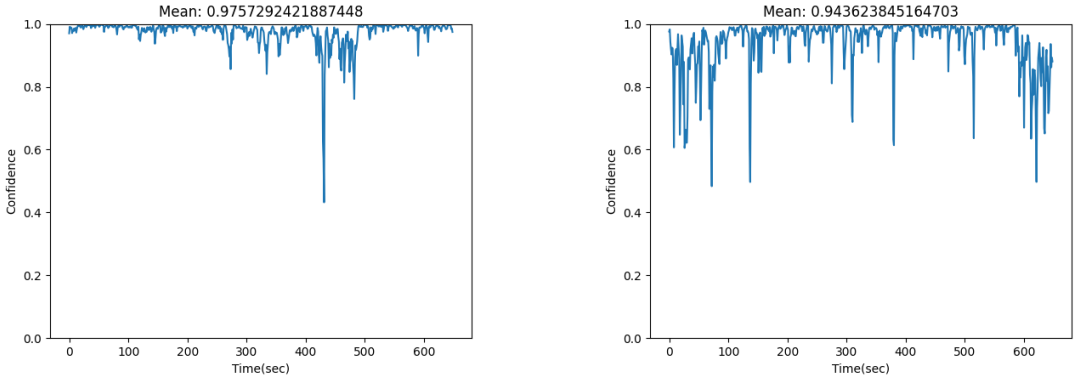
3.3.3 Confidence threshold

The confidence threshold is built to distinguish between known voice and unknown voice. According to Enrollment process figure, classifiers can return the result as predicted speaker and probability of that verdict. This probability is derived from votes of scheme of each (i, j) classifiers. By **interpreting the probability**, a suitable confidence threshold can be set for the system.



(a) Speaker 3

(b) Speaker 44



(c) Speaker 46

(d) Speaker 30

Figure 15: Confidence of 4 speakers in time domain

The above figures are taken from test set in enrollment process. Test audio length of each speaker is 600 seconds which equal 10 minutes. The system responses confidence score every 1 second from latest 3 seconds of speech which corresponding to 600 data points. Then these data points are connected then plotted to form to figures where the vertical axis represents confidence score and horizontal axis represents time domain in seconds.

From these figures, the range between the maximum and minimum confidence is approximately 0.6 where maximum is around 0.99 and minimum run from 0.3 to 0.5. The average confidence is relatively high at 0.93 to 0.97 when compare to the previous range of max and min values. This means the minimum values are only minority cases. Confidence threshold of 0.8 would cover the majority of cases.

4 Result and Discussion

4.1 Model test

The test goal is to examine the score of 4 classifiers trained by same machine learning method but different amount of data. Furthermore, the test is also check the capability to distinguish a voice in a set of known voices. The material use for this test is test voices set, derived from 46 speaker audio set. The test voices set has 10 minutes audio length for each classes which ensure balance of this test case.

The score of models are calculated using equation from section 1.3.4 Machine learning metrics. For multi-class classification, F1-score using macro approach, consider each class has same weight in average.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1 minute	0.95	0.95	0.95	0.95
5 minutes	0.97	0.97	0.97	0.97
10 minutes	0.99	0.99	0.99	0.99
20 minutes	0.99	0.99	0.99	0.99

Table 9: Score comparison of 4 classifiers

1 minute audio data after split to 3 seconds chunk with no overlap can have up to 20 audio chunks which correspond to 20 data point for the algorithm to draw decision boundaries. Although fit with only 20 data point (at least for each class) but 1 minute classifier has a really good score which about 0.95 accuracy. This means model can work well with just 1 minute audio. Thus, this also open an opportunity for the system to take live voice sample since it only take 1 minute of speech and another a few seconds to train the model.

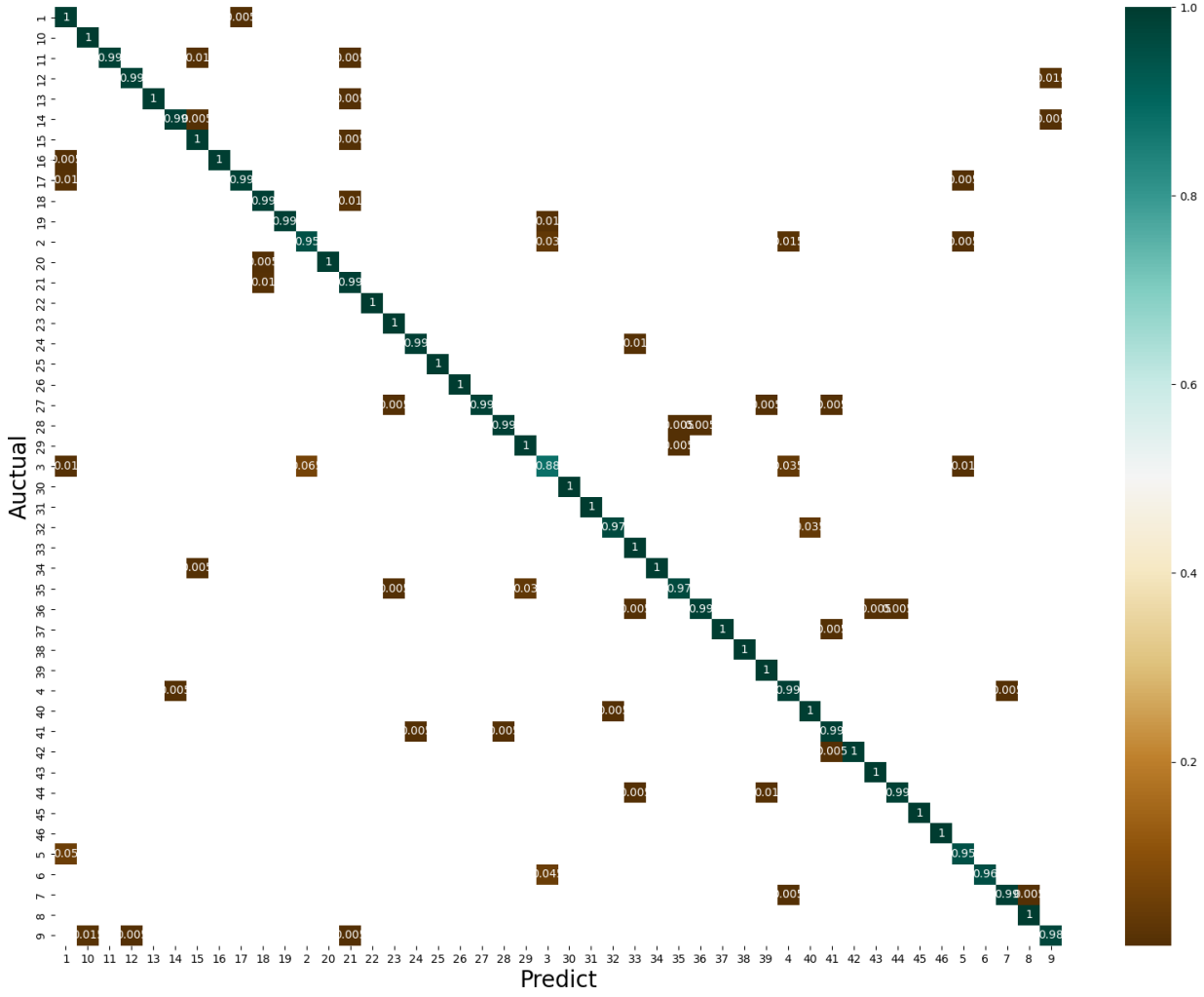


Figure 16: Confusion matrix of 1 minute classifier

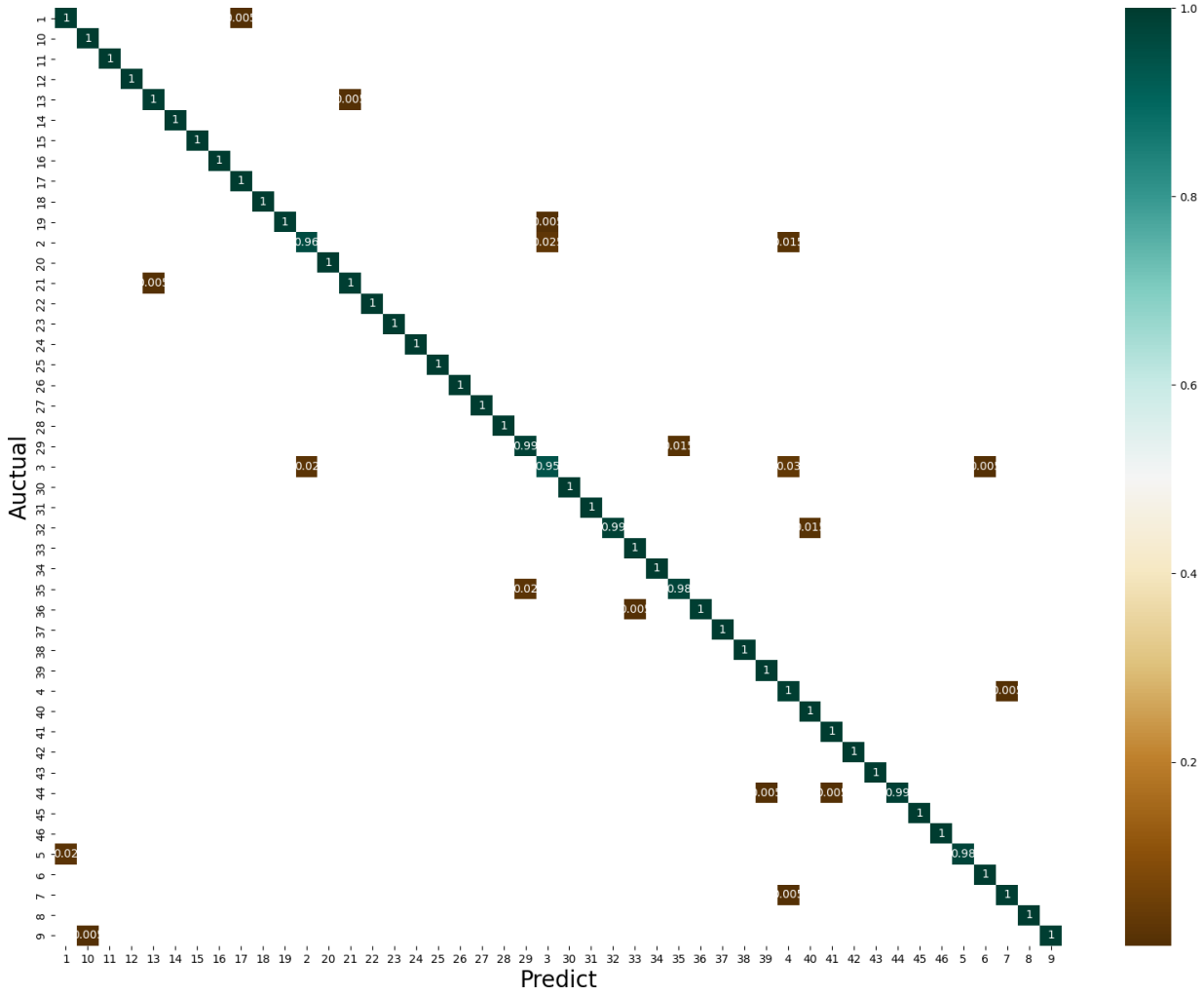


Figure 17: Confusion matrix of 20 minute classifier

Above figures are confusion matrix of 1 minute classifier and 20 minutes classifier. Visually, the two figures are modified for a better analysis by not color cases that are not exist. Generally, 20 minutes classifier has improve quite a lot when compare to 1 minute classifier. The frequency of false is gradually reduce.

4.2 Unknown voice test

The goal of this test is to determine whether the system can spot the known voice in a group of different voices. As description of figure 7, the unknown voice set and test set in enrollment process are used in this section as testing material. Unknown voice set consist over 16 minutes of speech with 6 different voice. To create a balance test case, 16 minutes of test set is added. In overall, after slitting into 3 seconds chunks, the total amount of sample is 648 samples including 320 samples are known voice and 328 samples are unknown voice. The test use 20 minutes classifier as classification model for system.

Metrics	**Value**
Accuracy	0.95
Precision	0.98
Recall	0.99
F1-Score	0.98

Table 10: Unknown voice test scores

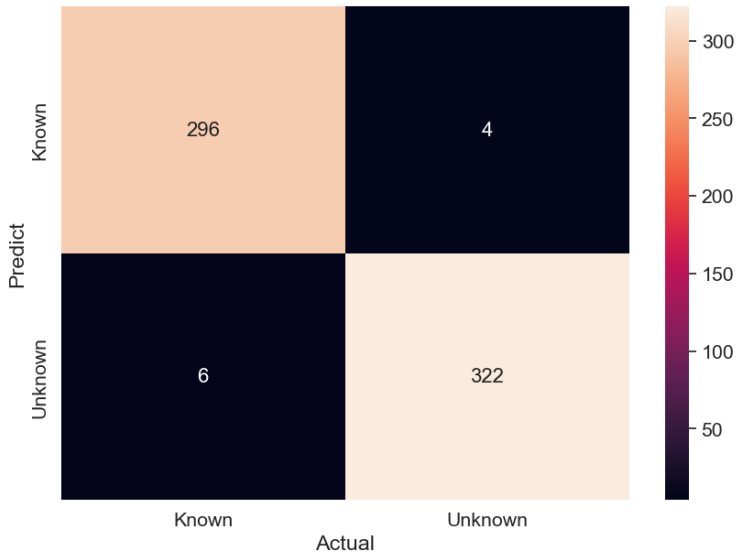


Figure 18: Unknown voice test confusion matrix

Achieving an accuracy score over 95%, the model has proven its ability to distinguish voices from speakers dataset and voices from other source that does not belong to any speakers in dataset.

4.3 Noisy voice test

The aim of this test is to know whether the system is able to recognize voice if it affect by environment noise. This test use noisy audio set as testing material and 20 minutes classifier as model for system. As previously stated in noisy audio set, 10 minutes voice is recorded. After that, 5 different noises is added to the 10 minutes length voice to create 5 version of voice which also means that each version would have a total of **600 data points**. A data point would called success if that data point pass two conditions: assigned as speaker 46 and has confidence score equal or over the confidence threshold (0.8).

Noises	**Average confidence score**	**Success cases**	**Accuracy(%)**
Dishwasher	0.64	79	13.16
Crushing leaves	0.43	13	2.16
Room	0.36	0	0
Rain	0.17	0	0
Theatre hall	0.66	114	19

Table 11: Noisy voice test result

As the result, it seems like the model meet a lot of difficult to recognize voice in noisy environment. In detailed, data points are assigned with multiple class instead of one true class. For room and rain noises, most of data point are not even pass the confidence threshold. Finally, theatre hall has most positive result out of all other noises. Although it has 19% of data point are pass the conditions but able to achieved 19% is also an effort for a noisy place like theatre hall.

4.4 Discussion

There are a few facts can be conclude after these tests. Firstly, model is sable to determine voice in a group of known voice. Additionally, 1 minute classifier show a great score despite only fit using minimum 1 minute audio. In fact, it raise many potential for future development for the project. Secondly, the system successfully distinguish between known voice and unknown voice with a well accuracy score. Thirdly, system does not work effectively when voice is affect by noises. This fact has proven that noise actually can change speech quality. Moreover, the impact of noise on speech could be different based on many factors such as, amplitude and frequency of that noise.

5 Conclusion and Future work

5.1 Conclusion

In conclusion, the project still has a lot of space for improvement, the system is far from completed product. Initially, it require a certain amount of basic knowledge about signal to actually hands-on this project since digital signal processing is not a new topic but difficult to dive deep insight. It took a while to start off from the fundamental of knowledge and learn enough for this project. Furthermore, real-time voice input implementation is generally in process because of time and the lack of skills to debug. The problem of real-time voice input come from many sources like library, unpack the string format of input voice and device difference between recorder and speaker despite of same audio settings. Also, the incapability of recognize known speech in noise environment is crucial. Although, there are many solutions to denoise, remove background noise but lack of time and ability to implement it to the system properly.

On the other hand, there are objectives that are successfully achieved. A text-independent speaker recognition system is made. The system is able to determine whose voice is in a set of known voices and distinguish the voice that is not belong to the set. And the system can give an accurate prediction in a second with only 3 seconds of speech as well. In addition, this project is also discover that 1 minute of speech when using Support vector classification is well enough. This was not expected since the sufficient data for training a speaker recognition model is usually above 40 minutes of speech[36].

5.2 Future work

As mentioned above, the project does not look completed yet. And these improvement would somewhat accomplished the project:

- Real-time voice input: The importance of real-time voice input is about making the speaker recognition system automatically. If the system successfully added this feature, the usability of it would increase dramatically.

- Audio filters: There are number of audio filter like using library, deep learning models. For simpler way, applying low-pass filter or high-pass filter can remove any frequency that too low or too high which is also extremely useful in noise reduction[37].

- Implementing on microcomputer: After the system is completed, pushing the program on other micro device like rasberry pi or arduino for further works. Adding this feature could increase scalability and adaptability of the project.

6 References

[1] Yugesh Verma. A tutorial on spectral feature extraction for audio analytics. URL https://analyticsindiamag.com/a-tutorial-on-spectral-feature-extraction-for-audio-analytics/.

[2] Samaneh Madanian, Talen Chen, Olayinka Adeleye, John Michael Templeton, Christian Poellabauer, Dave Parry, and Sandra L. Schneider. Speech emotion recognition using machine learning — a systematic review. *Intelligent Systems with Applications*, 20:200266, 2023. ISSN 2667-3053. doi: https://doi.org/10.1016/j.iswa.2023.200266. URL https://www.sciencedirect.com/science/article/pii/S2667305323000911.

[3] Wikipedia. Support vector machine. URL https://en.wikipedia.org/wiki/Support_vector_machine/.

[4] Rafizah Mohd Hanifa, Khalid Isa, and Shamsul Mohamad. A review on speaker recognition: Technology and challenges. *Computers Electrical Engineering*, 90:107005, 2021. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2021.107005. URL https://www.sciencedirect.com/science/article/pii/S0045790621000318.

[5] Jinxi Guo, Ning Xu, Kailun Qian, Yang Shi, Kaiyuan Xu, Yingnian Wu, and Abeer Alwan. Deep neural network based i-vector mapping for speaker verification using short utterances. *Speech Communication*, 105:92–102, 2018. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2018.10.004. URL https://www.sciencedirect.com/science/article/pii/S0167639318300360.

[6] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[7] M.H. Moattar and M.M. Homayounpour. A review on speaker diarization systems and approaches. *Speech Communication*, 54(10):1065–1103, 2012. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2012.05.002. URL https://www.sciencedirect.com/science/article/pii/S0167639312000696.

[8] Sadaoki Furui. Chapter 7 - speaker recognition in smart environments. In Hamid Aghajan, Ramón López-Cózar Delgado, and Juan Carlos Augusto, editors, *Human-Centric Interfaces for Ambient Intelligence*, pages 163–184. Aca-

demic Press, Oxford, 2010. ISBN 978-0-12-374708-2. doi: https://doi.org/10.1016/B978-0-12-374708-2.00007-3. URL https://www.sciencedirect.com/science/article/pii/B9780123747082000073.

[9] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li. Text-dependent speaker verification: Classifiers, databases and rsr2015. *Speech Communication*, 60:56–77, 2014. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2014.03.001. URL https://www.sciencedirect.com/science/article/pii/S0167639314000156.

[10] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52 (1):12–40, 2010. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2009.08.009. URL https://www.sciencedirect.com/science/article/pii/S0167639309001289.

[11] arvindpdmn Saranga-K-Mahanta-google, 2016. URL https://devopedia.org/audio-feature-extraction. "AudioFeatureExtraction.".

[12] Dan Ellis. Chroma feature analysis and synthesis. URL https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/.

[13] DUSTI MIRAGLIA. What is rms in audio? URL https://unison.audio/what-is-rms-in-audio/.

[14] Constantin Constantinescu and Remus Brad. An overview on sound features in time and frequency domain. *International Journal of Advanced Statistics and ITC for Economics and Life Sciences*, 13(1):45–58, 2023. doi: doi:10.2478/ijasitels-2023-0006. URL https://doi.org/10.2478/ijasitels-2023-0006.

[15] Alejandro A. Torres-García, Omar Mendoza-Montoya, Marta Molinas, Javier M. Antelis, Luis A. Moctezuma, and Tonatiuh Hernández-Del-Toro. Chapter 4 - pre-processing and feature extraction. In Alejandro A. Torres-García, Carlos A. Reyes-García, Luis Villaseñor-Pineda, and Omar Mendoza-Montoya, editors, *Biosignal Processing and Classification Using Computational Learning and Intelligence*, pages 59–91. Academic Press, 2022. ISBN 978-0-12-820125-1. doi: https://doi.org/10.1016/B978-0-12-820125-1.00014-2. URL https://www.sciencedirect.com/science/article/pii/B9780128201251000142.

[16] Bin Zhen, Xihong Wu, Zhimin Liu, and Huisheng Chi. On the importance of components of the mfcc in speech and speaker recognition. volume 37, pages 487–490, 10 2000. doi: 10.21437/ICSLP.2000-313.

[17] Mario Pisa. Gaussian distribution: What it is, how to calculate, and more. URL https://blog.quantinsti.com/gaussian-distribution/.

[18] Guillaume GUERARD. normalize-standardize-resize-your-data. URL https://complex-systems-ai.com/en/data-analysis/normalize-standardize-resize-your-data/#google_vignette.

[19] Ayushi Jain. Support vector machine (svm) algorithm. URL https://www.geeksforgeeks.org/support-vector-machine-algorithm/.

[20] Pawan Lingras and Cory Butz. Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification. *Information Sciences*, 177 (18):3782–3798, 2007. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2007.03.028. URL https://www.sciencedirect.com/science/article/pii/S0020025507001594.

[21] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *ArXiv*, abs/2008.05756, 2020. URL https://api.semanticscholar.org/CorpusID:221112671.

[22] Tavish Srivastava. Important model evaluation error metrics. URL https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/.

[23] Becky Scarrot. Mp3, aac, wav, flac: all the audio file formats explained. URL https://www.whathifi.com/advice/mp3-aac-wav-flac-all-the-audio-file-formats-explained.

[24] SciPy. read. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.read.html.

[25] Griffin Brown. Digital audio basics: Audio sample rate and bit depth. URL https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html.

[26] Alex Mixing Lessons. Best audio sample rate. URL https://www.mixinglessons.com/sample-rate/#:~:text=Sample%20rate%20refers%20to%20the%20number%20of%20samples,frequency%20range%20that%20can%20be%20captured%20and%20reproduced.

[27] Wildlife Aucostic. What is an audio channel. URL https://www.wildlifeacoustics.com/resources/faqs/what-is-an-audio-channel#:~:text=A%20digital%20audio%20file%20can%20contain%20multiple%20channels,movie%20audio%20is%20often%20mixed%20for%206%20channels.

[28] VIBHOR JAIN. Speaker recognition audio dataset. URL https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset.

[29] Librvox. Free public domain audiobookst. URL https://librivox.org.

[30] edureka! Education youtube channel. URL https://www.youtube.com/@edurekaIN.

[31] GSMArena. Xiaomi redmi10. URL https://www.gsmarena.com/xiaomi_redmi_10-11060.php.

[32] Sample focus. Noise. URL https://samplefocus.com/tag/noise.

[33] Eric Ravenscraft. What is 32-bit float audio, and should you record in it. URL https://www.wired.com/story/32-bit-float-audio-explained/.

[34] A. Aylin Tokuç. Normalization vs standardization in linear regression. URL https://www.baeldung.com/cs/normalization-vs-standardization.

[35] Phorce. What are chunk when recording a voice signal. URL https://dsp.stackexchange.com/questions/13728/what-are-chunks-when-recording-a-voice-signal.

[36] nitinme eric urban. What is speaker recognition? URL https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speaker-recognition-overview.

[37] Rupert Monk. High pass vs low pass filter: Understanding audio frequency filters. URL https://musicalstudy.com/high-pass-vs-low-pass-filters/#:~:text=High%20pass%20and%20low%20pass%20filters%20are%20two,point%20to%20pass%20through%2C%20effectively%20removing%20lower%20frequencies.

7 Appendix

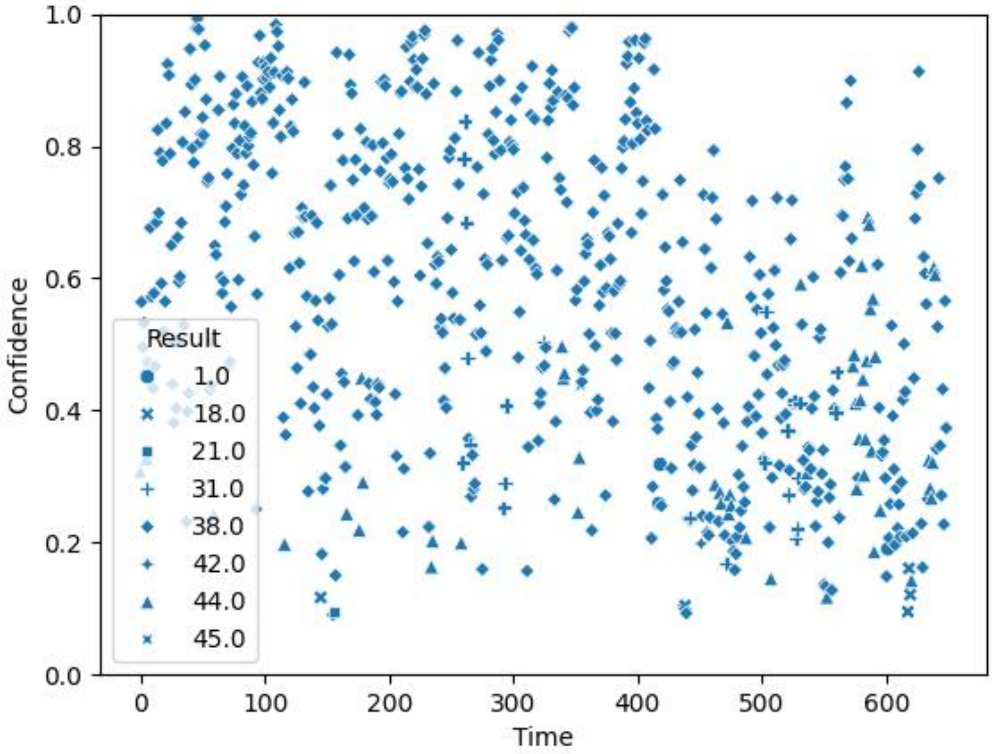


Figure 19: Voice in dishwasher noise confidence on time domain

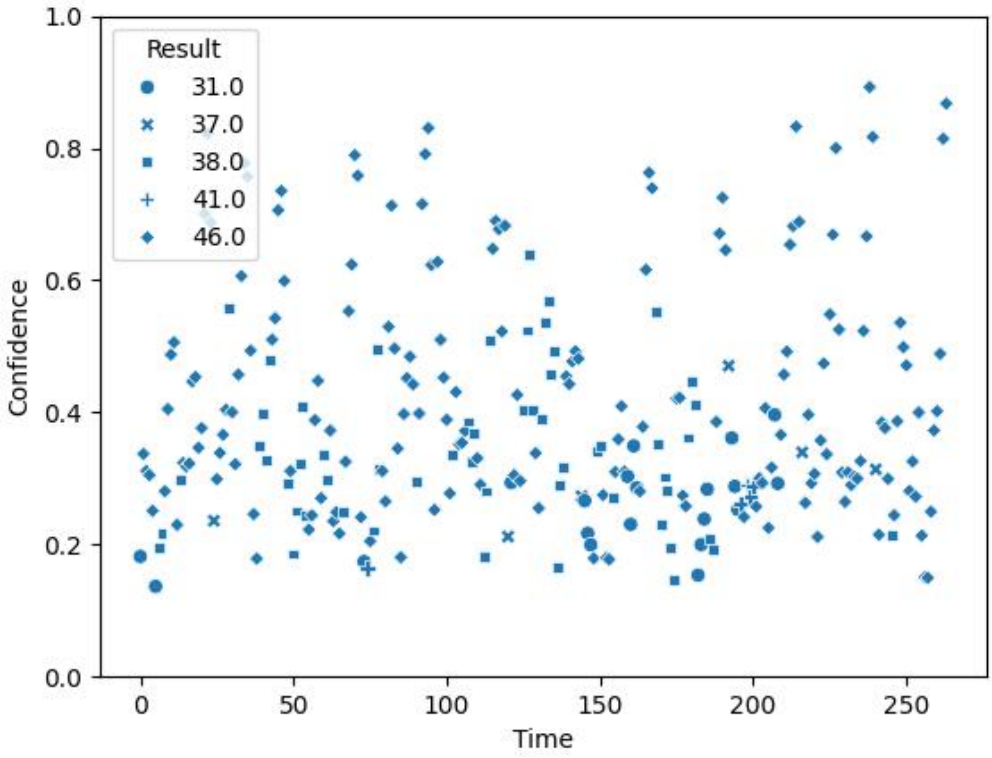


Figure 20: Voice in crushing leaves noise confidence on time domain

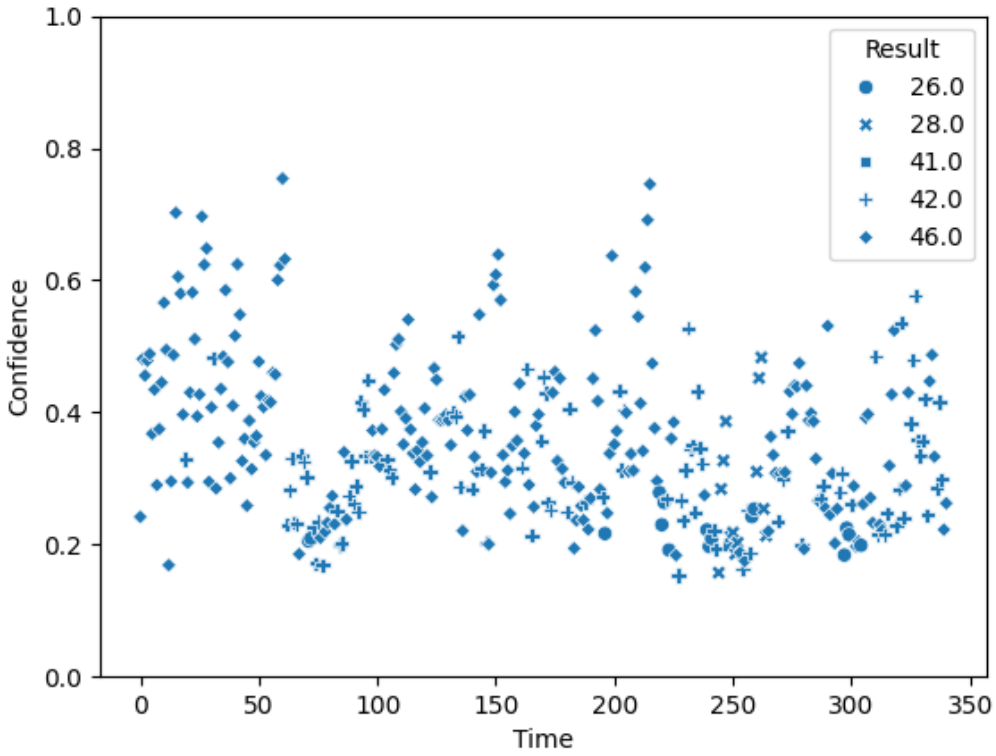


Figure 21: Voice in room noise confidence on time domain

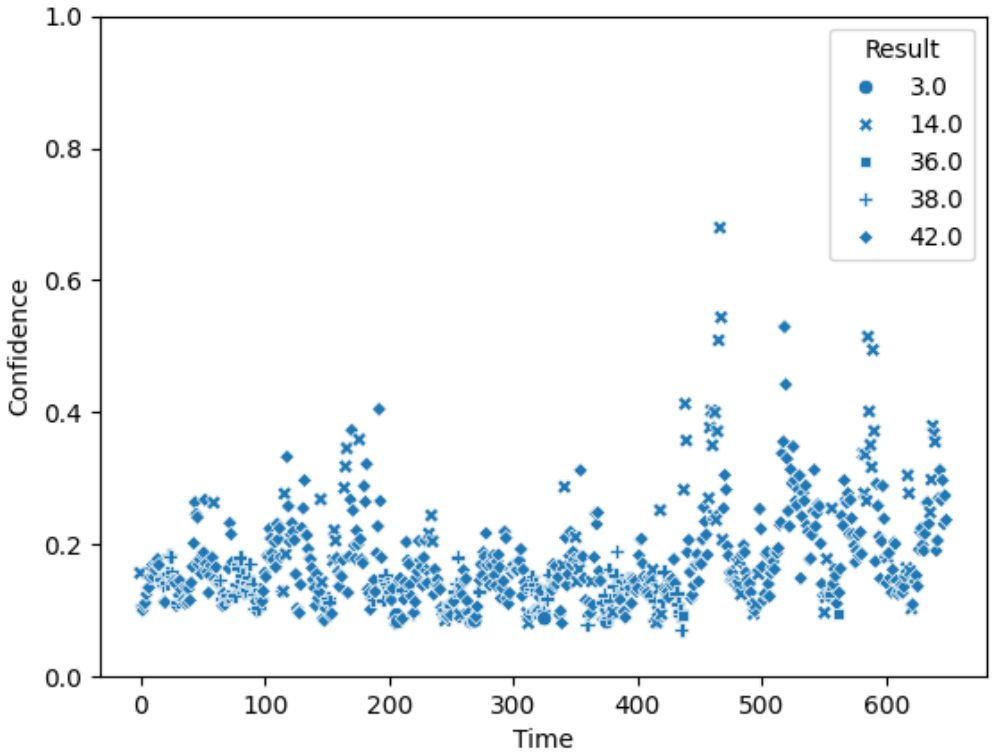


Figure 22: Voice in rain noise confidence on time domain

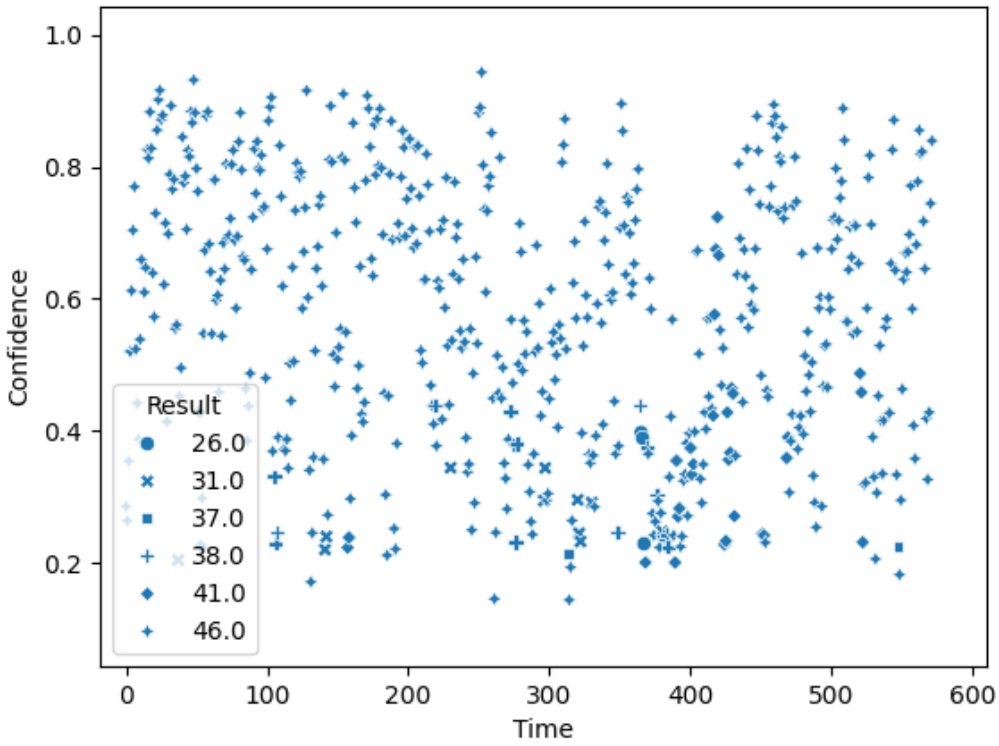


Figure 23: Voice in theatre hall noise confidence on time domain

The end!