Qatar University

Department of Computer Science and Engineering

Course Project - CMPE 476

Model Development for Pre-Recorded Voice Commands for Automated Home

Project Group Member:

Hagr Abdelhamid (201902720 – Section L52)
Marim Abdelhamid (201902721 – Section L52)
Marim Elhanafy (201803468 – Section L52)

Date: 08/06/2023

.

Abstract

As technology advances, automated homes are becoming more popular, providing homeowners with convenience and efficiency. Voice-controlled systems are key to interacting with these automated devices smoothly. However, existing voice command recognition systems are designed for real-time speech input and may not work well with pre-recorded voice commands. This project aims to create a strong model that can accurately understand and respond to pre-recorded voice commands in automated homes. This project aims to improve the recognition of pre-recorded voice commands in automated homes by using a special filter called the FIR filter. The FIR filter will help make the recorded voice commands clearer and reduce any unwanted noise. By doing this, the project hopes to make the voice commands more accurate and dependable for controlling automated home systems. The project will involve designing and applying the FIR filter to the recorded voice commands, and then testing to see if it improves the performance of the automated home system based on MSE (Mean Squared Error). The two prerecorded voice commands that we will use in this project are "AC" and "Low." The data set will consist of twenty files for each word that have been recorded by males and females.

Table of Contents

1. Project Description

This project addresses the challenge of achieving precise recognition and appropriate response to pre-recorded voice commands within automated home environments. Voice command recognition in automated homes is useful. It lets the users control devices like lights, thermostats, security systems, and appliances just by using their voices. They can also automate tasks with personalized voice commands, making things more convenient. It is great for people who have trouble moving around because they can control their home with their voice. It is hands-free and can be customized upon their preferences. They can even connect it to virtual assistants like Alexa or Google Assistant to do even more. Overall, it makes life easier and more enjoyable in the automated home. The technical challenges of the project include that the system may not be fully accurate, making it difficult to distinguish similar words based on their sounds. Another challenge is that some people experience discomfort when pronouncing certain letters or words. To address these challenges, we will create a model that assigns coefficients to each word. We will then evaluate other words by multiplying their coefficients with our model to determine if they match. Our goal is to build an audio recognition system that can accurately identify and differentiate between words, such as “AC” and "LOW". By accomplishing this project, we aim to provide homeowners with an accurate and reliable system for interacting with their automated homes using pre-recorded voice commands. This will enhance the usability, convenience, and overall user experience.

2. Project Design

2.1. Overall Block Diagram

The overall Model Order (N) Selection process for the Best Model is shown in Figure 1. The data collection is followed by selecting the model order (N) and reading the audio data. Preprocessing involves selecting the first channel of each audio and reducing sample numbers using discrete wavelet transform (dwt). Then, the Y and Sai matrices are constructed. Then, calculating the theta from Y and Sai. After that, calculating Y-hat. Finally, calculating the mean squared error to determine the best (N) for the model. Figure 2 shows the workflow of the training part which includes file reading, preprocessing, building Y and Sai, and calculating theta. Figure 3 shows the workflow of the training part which includes file reading, preprocessing, building Y and Sai, and calculating Y-hat. Then, calculating the mean squared error (MSE) and check if it is less than 10% that is mean the detected word presents the actual word and vice versa.

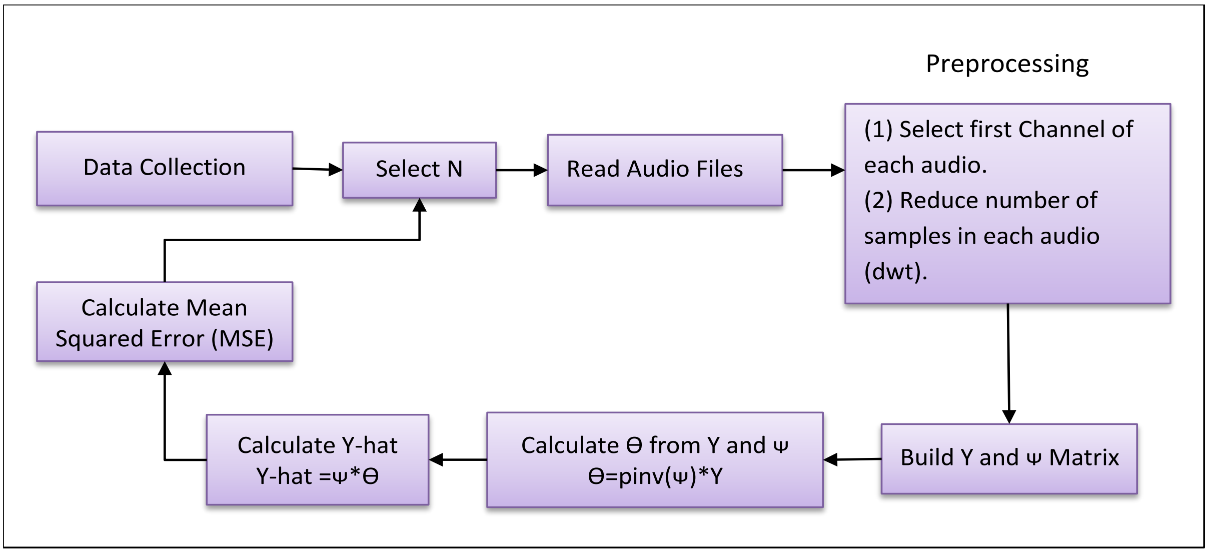


Figure 1. Model Order (N) Selection for the Best Model

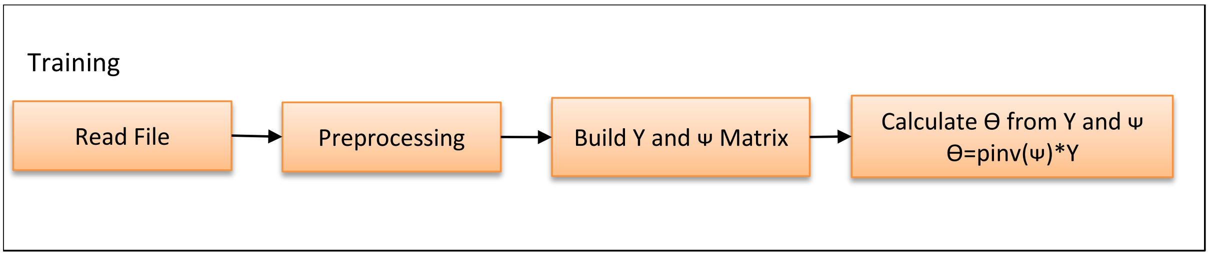


Figure 2. Training

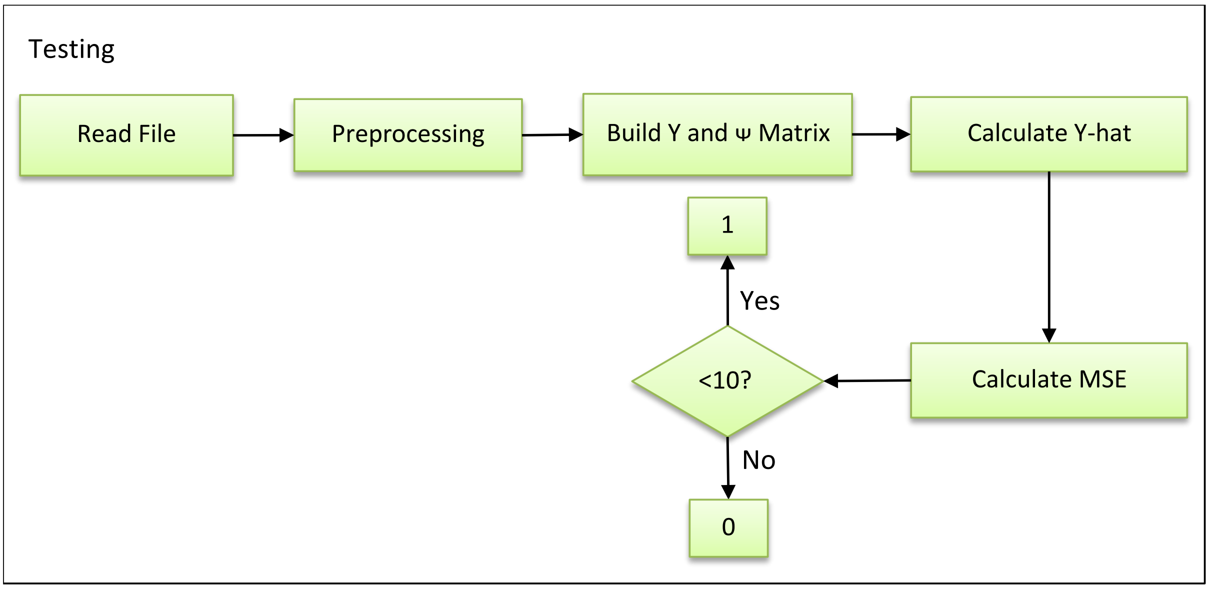


Figure 3. Testing

2.2. Model Structure

In our project, the autoregressive model is used to predict current values of the audio data based on its previous values. The input audio in m4a format is first transformed using discrete wavelet transform (dwt) to reduce the number of samples. Then, an autoregressive model is applied by estimating autoregressive coefficients, which capture the influence of previous values on the current value. These coefficients are used to calculate predicted values. Finally, the mean square error (MSE) is calculated to evaluate the accuracy of the predictions. The autoregressive model helps capture temporal dependencies and make predictions based on the audio's past behavior.

In the autoregressive model, the estimated autoregressive coefficients θ are typically obtained through a fitting process using the following equation:

$$\theta = \psi^{-1} * Y$$

These coefficients are then used in the autoregressive model equation to predict the current values. The formulation of the autoregressive model equation would be:

$$\hat{Y} = \psi * \theta$$

In this equation, $\hat{Y}$ represents predicted values, θ are the autoregressive coefficients, and ψ represents lagged values. To calculate $\hat{Y}$ based on the autoregressive model equation, we would iterate through the data, utilizing the previous values ψ and autoregressive coefficients θ to predict the current value $\hat{Y}$. These are typical values iteratively, starting from the initial lagged values, and proceeding until the prediction is reached (based on model order N).

2.3. Modified Code

Here is the enhanced version of the code that was done on the day of Projecthon. The code is divided into 3 files. IsAC function to decide if the selected file represents the word AC or not. In addition, IsLow function to decide if the selected file represents the word Low or not. The last file is used for testing which is used to call both IsAc and IsLow functions. These files are represented as follows:

IsAC function

```
function result = isAC(filename)
%isAC function to check the audiofile
%   if the audio represent AC word, it will return true = 1
%   else it will return False = 0
M = [];
for N = 100:10:150
    Y = [];
    Sai = [];
    theta = [];
    s = 800;
    for i = 1:10
        name =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanaf
y_MarimAbdelhamid_HagrAbdelhamid\AC\AC_train\AC_' num2str(i) '.m4a'];
        [y, Fs] = audioread(name);
        y = y(:,1);
        y = detrend(y);
        [CA, CD] = dwt(y,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        for j = N+1:s
            Y = [Y; CA(j)];
            Sai = [Sai; [CA(j-1:-1:j-N)']];
        end
    end
    theta = pinv(Sai)*Y;
    Y_hat = Sai*theta;
    M = [M; mse(Y,Y_hat)];
end
N = 100:10:150;
disp('First: the best N is selected based on the Mean Square Error');
disp('As shown in the figure below');
figure, plot(N,M);
title('MSE Vs. N');
xlabel('N'); ylabel('MSE');
grid on; grid minor;
disp('Second: The best N is 110');
% Evaluate Y for the test audiofile
N = 110;
Sai_train = [];
Y_train = [];
theta_N = [];
name =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanaf
y_MarimAbdelhamid_HagrAbdelhamid\AC\AC_train\AC_' num2str(2) '.m4a'];
[y, Fs] = audioread(name);
y = y(:,1);
y = detrend(y);
[CA, CD] = dwt(y,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
for j = N+1:s
    Y_train = [Y_train; CA(j)];
    Sai_train = [Sai_train; CA(j-1:-1:j-N)'];
end
theta_N = pinv(Sai_train)*Y_train;

Sai_test = [];
[y, Fs] = audioread(filename);
y = y(:,1);
y = detrend(y);
[CA, CD] = dwt(y,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
for j = N+1:s
    Sai_test = [Sai_test; CA(j-1:-1:j-N)'];
end
Y_hat = Sai_test*theta_N;
M = mse(Y_train,Y_hat);
disp(['The MSE of the audiofile selected at (N=110) = ' num2str(M)]);
if M <= 0.1
    result = 1;
else
    result = 0;
end
end
```

IsLow function

```
function result = isLow(filename)
%isLow function to check the audiofile
%   if the audio represent Low word, it will return true = 1
%   else it will return False = 0
M = [];
for N = 80:10:130
    Y = [];
    Sai = [];
    theta = [];
    s = 800;
    for i = 1:10
        name =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanaf
y_MarimAbdelhamid_HagrAbdelhamid\AC\AC_train\AC_' num2str(i) '.m4a'];
        [y, Fs] = audioread(name);
        y = y(:,1);
        y = detrend(y);
        [CA, CD] = dwt(y,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        [CA, CD] = dwt(CA,'sym2');
        for j = N+1:s
            Y = [Y; CA(j)];
            Sai = [Sai; [CA(j-1:-1:j-N)']];
        end
    end
    theta = pinv(Sai)*Y;
    Y_hat = Sai*theta;
    M = [M; mse(Y,Y_hat)];
end
N = 80:10:130;
disp('First: the best N is selected based on the Mean Square Error');
disp('As shown in the figure below');
figure, plot(N,M);
title('MSE Vs. N');
xlabel('N'); ylabel('MSE');
grid on; grid minor;
disp('Second: The best N is 110');
% Evaluate Y for the test audiofile
N = 110;
Sai_train = [];
Y_train = [];
theta_N = [];
name =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanaf
y_MarimAbdelhamid_HagrAbdelhamid\Low\Low_train\Low_' num2str(2) '.m4a'];
[y, Fs] = audioread(name);
y = y(:,1);
y = detrend(y);
[CA, CD] = dwt(y,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
for j = N+1:s
    Y_train = [Y_train; CA(j)];
    Sai_train = [Sai_train; CA(j-1:-1:j-N)'];
```

```
end
theta_N = pinv(Sai_train)*Y_train;

Sai_test = [];
[y, Fs] = audioread(filename);
y = y(:,1);
y = detrend(y);
[CA, CD] = dwt(y,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
[CA, CD] = dwt(CA,'sym2');
for j = N+1:s
    Sai_test = [Sai_test; CA(j-1:-1:j-N)'];
end
Y_hat = Sai_test*theta_N;
M = mse(Y_train,Y_hat);
disp(['The MSE of the audiofile selected at (N=110) = ' num2str(M)]);
if M <= 0.1
    result = 1;
else
    result = 0;
end
end
```

Testing file

```
clear all; close all; clc;
% Test AC
TP = 0;
FP = 0;
TN = 0;
FN = 0;
for i = 1:10
name1 =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\Test\Test_' num2str(i) '.m4a'];
[result] = isAC(name1);
fprintf('The result is %d \n',result);
    if contains(name1, 'AC')
        if result == 1
            TP = TP + 1;
        else
            FP = FP + 1;
        end
    else
        if result == 1
            FN = FN + 1;
        else
            TN = TN + 1;
        end

    end
end
fprintf('TP is  %d \n',TP);
fprintf('FP is  %d \n',FP);
fprintf('TN is  %d \n',TN);
fprintf('FN is  %d \n',FN);
% Test Low
TP = 0;
FP = 0;
```

```
TN = 0;
FN = 0;
for i = 1:10
name2 =
['C:\Users\97455\Desktop\MarimElhanafy_MarimAbdelhamid_HagrAbdelhamid\MarimElhanaf
y_MarimAbdelhamid_HagrAbdelhamid\Test\Test_' num2str(i) '.m4a'];
[result] = isLow(name2);
fprintf('The result is %d \n',result);
    if contains(name2, 'Low')
        if result == 1
            TP = TP + 1;
        else
            FP = FP + 1;
        end
    else
        if result == 1
            FN = FN + 1;
        else
            TN = TN + 1;
        end

    end
end
fprintf('TP is  %d \n',TP);
fprintf('FP is  %d \n',FP);
fprintf('TN is  %d \n',TN);
fprintf('FN is  %d \n',FN);
```

The code is enhanced as a different type of wavelet is used to reduce the number of samples in each audio file (sym2 instead of db2). Moreover, normalization is removed as it increases mean square error. Furthermore, model order N that represents the best model is updated. In addition, calculation of True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) is added in the testing file to calculate accuracy, sensitivity, specificity for each model built.

3. Testing and Results

At the beginning of each function (IsAC and IsLow), The first step was to get the model order N that will lead to the best model. That is done by calculating mean square error (MSE) for each N selected. Then, plot MSE with respect to selected N to choose the best N which correspond to lowest mean square error (MSE). The MSE vs. N plots for both AC and Low models are shown below:

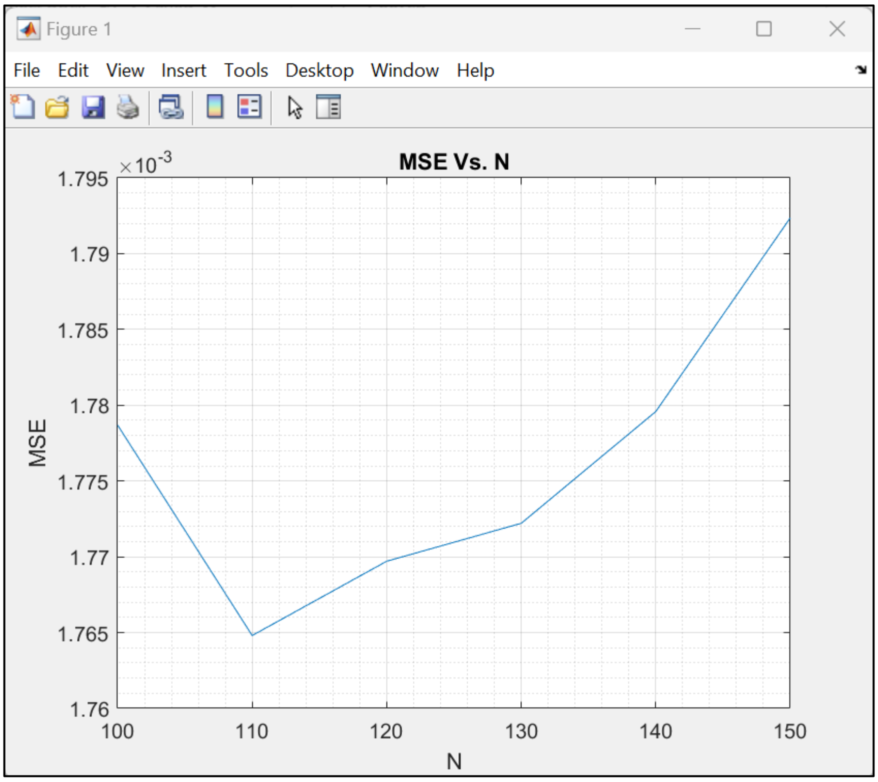


Figure 4. MSE Vs. N plot for AC model

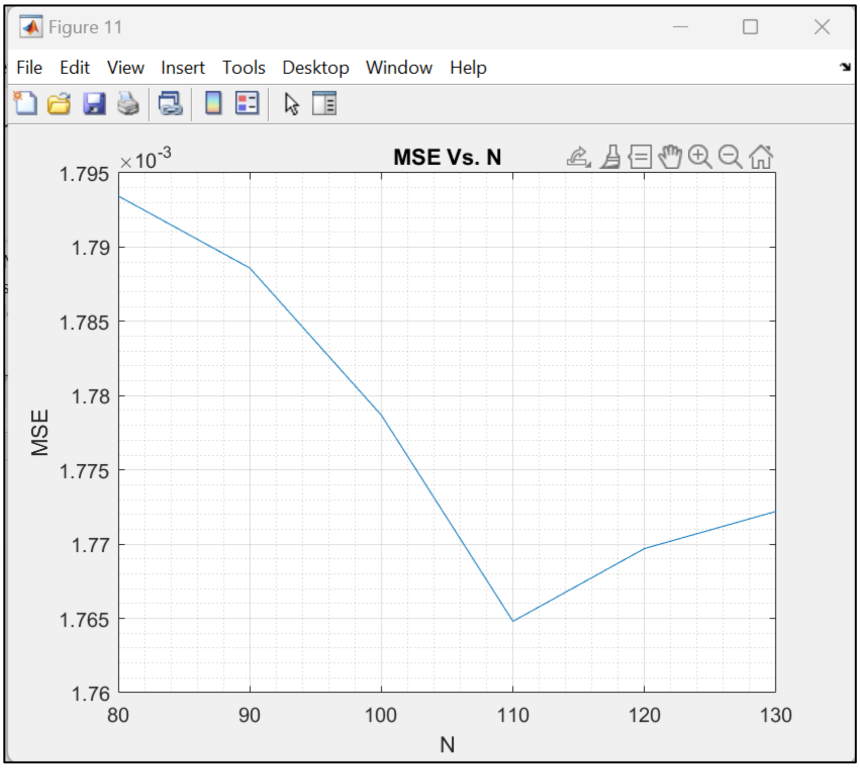


Figure 5. MSE Vs. N plot for Low model

As shown in Figures 4 and 5, the model order that shows the least MSE is 110 for both models. This order is used for both training and testing stages.

Both models are tested using 10 different files that represent the actual word (AC words are input to AC model, and Low words are input to Low model). The output after testing was as following:

```
First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.060944

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.029782

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.08265

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0089869

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.024325

The result is 1
```

```
First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.010281

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.10991

The result is 0

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.10109

The result is 0

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.1165

The result is 0

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0086618

The result is 1

TP is  7

FP is  3

TN is  0
```

```
FN is  0

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.043608

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.019668

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.011532

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0055944

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0055949

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below
```

```
Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0055954

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.037961

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.041685

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.043608

The result is 1

First: the best N is selected based on the Mean Square Error

As shown in the figure below

Second: The best N is 110

The MSE of the audiofile selected at (N=110) = 0.0056178

The result is 1

TP is  10

FP is  0

TN is  0

FN is  0
```

The first 10 files are input to AC model and second 10 files are the input to Low model. From the above output, Accuracy of each model can be calculated as follows:

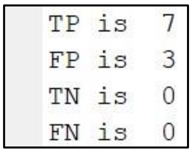
$$Accuracy = \frac{Correctly\ detected}{Total\ number\ of\ files} * 100\% = 70\%$$

$$Accuracy_{AC} = \frac{7}{10} * 100\% = 70\%$$

$$Accuracy_{Low} = \frac{10}{10} * 100\% = 100\%$$

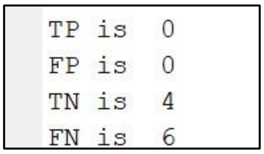
In Addition, True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) were calculated after testing each model using 20 files. The output was as following:

1. After testing AC model using 10 AC files



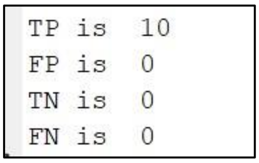
```
TP is  7
FP is  3
TN is  0
FN is  0
```

2. After testing AC model using 10 non-AC files



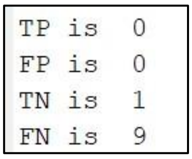
```
TP is  0
FP is  0
TN is  4
FN is  6
```

3. After testing Low model using 10 Low files



```
TP is  10
FP is  0
TN is  0
FN is  0
```

4. After testing Low model using 10 non-Low files



```
TP is  0
FP is  0
TN is  1
FN is  9
```

From the above output, Sensitivity, Specificity, and Accuracy can be calculated for each model as following:

AC model:

$$Sensitivity_{AC} = \frac{TP}{TP + FN} * 100\% = \frac{7}{7+6} * 100\% = 53.85\%$$

$$Specificity_{AC} = \frac{TN}{TN + FP} * 100\% = \frac{4}{4+3} * 100\% = 57.14\%$$

$$Accuracy_{AC} = \frac{TN + TP}{TP + TN + FP + FN} * 100\% = \frac{4+7}{7+4+3+6} * 100\% = 55\%$$

Low model:

$$Sensitivity_{Low} = \frac{TP}{TP + FN} * 100\% = \frac{10}{10 + 9} * 100\% = 52.63\%$$

$$Specificity_{Low} = \frac{TN}{TN + FP} * 100\% = \frac{1}{1 + 0} * 100\% = 100\%$$

$$Accuracy_{Low} = \frac{TN + TP}{TP + TN + FP + FN} * 100\% = \frac{1 + 10}{10 + 1 + 0 + 9} * 100\% = 55\%$$