International Journal of Informatics and Computation (IJICOM)
Vol. 7, No. 1, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Speaker Recognition System Using MFCC and HMM Methods

Sahid Sultoni[1], Budi Darmawan[2], Supriono[3]

Abstract

Speaker recognition is a technology used to identify a person's identity based on their voice characteristics. This research aims to develop a speaker recognition system using the Mel Frequency Cepstral Coefficients (MFCC) method for voice feature extraction and the Hidden Markov Model (HMM) for speaker classification. Voice data was collected from 30 speakers in a total of 1500 voice samples. We test the data using the HMM model with five state configurations after preprocessing and feature extraction using MFCC. The test results showed that the accuracy of the training data ranged from 89.50% to 95.67%, while the accuracy of the test data was in the range of 83.63% to 89.46%. By conducting rigorous evaluation, it can demonstrate superior performance in recognizing speakers with a high degree of accuracy with a combination of MFCC and HMM.

Keywords:
Speaker recognition, MFCC, HMM, Feature Extraction, Speech Classification

1. Introduction

Speaker recognition is the process of recognizing the voice of a speaker. It is widely applied in various fields of life, such as remotely retrieving access to personal devices, securing access to voice control, and conducting forensic investigations. In speaker recognition, the most important process is to extract features from the speaker's voice. The features are used to represent the speech as unique features to distinguish speech samples from each other [1]. The speaker's voice has complex information in the field of speech recognition, as well as in the biometric field that analyzes human physical and behavior that aims for the authentication process [2]. The process of processing voice signals can use digital signal processing and certain algorithms that can process both mathematical functions or equations so that they can be recognized [3].

Voice recognition systems require feature extraction methods, one of the feature extraction methods is the Mel Frequency Cepstral Coefficients method. The Mel Frequency Cepstral Coefficients method is a sound feature extraction method that adopts the principle of human hearing senses with the aim of getting results that are as similar as possible to the human sense of hearing. This method starts from the pre-emphasis stage, frame blocking, windowing, Fast Fourier Transform, Mel frequency wrapping and cepstrum [4]. The MFCC feature extraction method is a noise-sensitive sound signal extraction method. The MFCC method produces high accuracy when in a clean environment. Conversely, when in a noisy environment the resulting accuracy drops dramatically [5].

HMM is an extension of the Markov chain where the state cannot be observed directly (hidden), but can only be observed through a set of other observations [6]. One of the voice recognition structure formers in order to work on the device is the Hidden Markov Model

Corresponding Author:1. Sahid Sultoni, University of Mataram, (sultoni1802@gmail.com)
2. Budi Darmawan, University of Mataram, budidarmawan@unram.ac.id
3. Supriono, University of Mataram, supriono@unram.ac.id

(HMM) voice recognition statistical model. The application of HMM in various cases shows that this model is suitable for a variety of data [7]. Hidden Markov Model (HMM) consists of a Markov chain in the first part that hides the state therefore the internal behavior of the model remains invisible. The hidden states of the model capture the temporal structure of the data. HMM is a statistical model that describes a sequence of events [8]. With the HMM method, we can obtain hidden parameters that can be used for further analysis. The method for HMM feature matching uses observations and states [3].

To improve the effectiveness of speaker recognition, this paper proposes an MFCC algorithm for speech feature extraction and applying HMM for speaker modeling and classification. In addition, the use of MATLAB as a development platform is expected to speed up the research process and facilitate the analysis and visualization of results.

2. Related Works

A study presented the Wavelet-MFCC Method and HMM to construct a character coefficient formation algorithm. The study aims to test and select the best channel from the wavelet-MFCC process as a new character coefficient for the speaker recognition system. The Wavelet- MFCC feature coefficient found that the detail channel (cD) as a feature can provide the same accuracy as using the combined channel (cAcD) and is higher than the approximation channel (cA), with an accuracy of 95% [8]. Another research proposed MFCC for a Voice Recognition System that starts from pre-emphasis, frame blocking, windowing, fast Fourier transform, mel frequency wrapping and cepstrum stages. Based on the test results, the MFCC in ideal condition reached 90% and the percentage of system failure was 10% with a top 5 error rate of 0%. Testing with non-ideal conditions, the percentage of system success was 76.6667% and the percentage of system failure was 23.333% with a top 5 error rate of 0% [4].

Another work explored MFCC and DTW to recognize the types of male and female voices. The results obtained are for the accuracy rate in women with alto voice type obtained 80% percentage, for mezzosopran voice obtained 90%, for soprano voice obtained 80% [9]. A study presented MFCC and VQ for gender recognition of speakers with text-dependent and speaker-dependent speech. In the paper, MFCC proceeds for feature extraction from speech signals while the clustering process uses the Vector Quantization (VQ) method. In the recognition stage, the paper adopted a distortion measure based on Euclidean distance minimization to match the test speakers with the speakers in the database. The speech database used 20 speakers, consisting of 10 male speakers and 10 female speakers with an accuracy rate of 90% for males and 80% for females [10].

Reading the holy Qur'an also utilized the HMM algorithm to recognize and pronounce the Hijaiyah Letters. The research results show that the test result of Hijaiyah letters at the same accuracy level is 100%, while the test of different letters is 54.6%. Thus, this research can help students to recognize and pronounce the Qur'an [11]. Another research proposed HMM for speech recognition model that shows the success rate of HMM in recognizing data reaches 71.43% [7]. Another work combined MFCC, Wavelet, and HMM to construct a Syllable-Based Indonesian Speech Recognition System. In this research, the process of syllable sound feature extraction applied MFCC and WPT methods. Recognition results with training data show the best accuracy of 100% for the WPT method and 75% for the MFCC method. While using test data, the best accuracy results are 100% for WPT db7, 83.33% for WPT db3, and 50% for MFCC. All of these best recognition results were obtained at the cut-off point of consonant sample length of 1024 samples [12].

3. Proposed Method

In this research, we utilize the MFCC and HMM methods for speaker recognition. The MFCC method is used for voice feature extraction and the HMM method is used for voice data classification. The combination of these two methods is done for the speaker recognition system. The stages of this research can be seen in Figure 3.1 below.

3.1 MFCC

MFCC is one of the feature extraction methods used in the field of speech processing. This method is used for a process that converts speech signals into several parameters. Extracting the best parametric representation of acoustic signals is an important task to produce better recognition performance. The efficiency of this stage is important for the next stage because it affects its behavior. MFCC is based on the perception of human hearing that cannot hear sounds with frequencies above 1 kHz in other words, in MFCC it is based on the known variation of the human ear's critical bandwidth with frequency. MFCC has two types of filters that are linearly spaced at low frequencies below 1000 Hz and logarithmic above 1000 Hz. The final result of the MFCC process is to get the cepstrum value. Cepstrum is the inverse fourier transform of the energy spectrum [13].

A. Pre-Emphasis

Pre-emphasis is the initial stage in the MFCC process. This stage is carried out because the signal often experiences noise interference, so it is necessary to reduce noise. The problem of noise in a very simple way is by filtering, namely pre-emphasis. Pre-emphasis aims to ensure that the baseband level in the high-frequency section still has good signal quality. The pre-emphasis process with an α value between 0 and 1 or between 0.9 ≤ α ≤ 1.0 uses Equation 1 [14].

$$y(n) = s(n) - a\, s(n-1) \quad (1)$$

Table 1. The Mathematical Notation of Input Gate

Notation	Description
$y(n)$	$Pre-emphasis\ result\ signal$
$s(n)$	Signal before pre-emphasis
a	pre-emphasis filter constant (between 0.9-1.0)

Because the sound signal is constantly changing due to the shifting of the articulation of the vocal production organs, the signal must be processed in short segments (short frames). The frame length that is usually used for signal processing is between 10-30 milliseconds. The length of the frame used greatly affects the success of spectral analysis. On the one hand, the size of the frame must be as long as possible to be able to show good frequency resolution. On the other hand, the frame size must also be short enough to be able to show good time resolution [15].

B. Windowing

The window function used is the one with a maximum value of 1 for the area inside the window and zero for the other areas. The window moves along the sound signal and extracts the signal shape inside it. (Hidayat et al., 2015) The framing process can cause spectral leakage or aliasing. This effect can occur due to the low number of sampling rates, or because of the frame-blocking process which causes the signal to become discontinuous. To reduce the possibility of spectral leakage, the results of the framing process must go through the windowing process. There are many window functions, w(n),

such as a good window function must narrow in the main lobe, and widen in the side lobe. Formula 2 shows the representation of the window function to the input sound signal [15].

$$x(n) = f_1\,(n)w(n) \tag{2}$$

Table 2. The Mathematical Notation of Input Gate

Notation	Description
$x(n)$	Windowing result signal
f_1	Frame blocking results (with n is 0.1, …, N-1)
N	Number of samples in each frame
$w(n)$	window function

C. Fast Fourier Transform (FFT)

Fourier transform is performed for each frame that has been formed. The goal is to change the signal from the time domain to the frequency domain. Because the result of this Fourier transform is symmetrical, only half of the transform result is taken. After that, this result is multiplied by its conjugate to obtain the signal spectrum.[16]. Fast Fourier Transform aims to decompose the signal into a sinusoidal signal in the form of real units and imaginary units. Fast Fourier Transform uses Equation 3 [14].
(2.3)

$$T(k) = \sum_{n=0}^{N-1} X(n)\cos\left(\frac{2\pi kn}{N}\right) - \sum_{n=0}^{N-1} X(n)\sin\left(\frac{2\pi kn}{N}\right) \tag{3}$$

Table 3. The Mathematical Notation of Input Gate

Notation	Description
$T(k)$	The result of the kth FFT calculation
$X(n)$	Results of the nth windowing calculation
k	index of frequency (1, 2, …, N)

FFT is one of the fastest algorithm methods to be able to implement Discrete Fourier Transform (DFT). DFT is a computational tool that plays a very important role in many digital signal processing applications, such as frequency analysis, power spectrum estimation, and linear filters. DFT computation time is too long and inefficient then FFT can perform calculations efficiently. FFT is used as an efficient way to be able to calculate DFT. Discrete Fourier Transform (DFT) uses Equation 4 [14].

$$d[k] = \sum_{n=0}^{N-1} X(n)e^{-j\frac{2\pi}{N}nk}; k = 0,1,2,...,N-1. \tag{4}$$

Table 4. The Mathematical Notation of Input Gate

Notation	Description
$d[k]$	DFT calculation results
$X(n)$	Windowing results
N	Number of samples to be processed
k	The discrete frequency variable has a value (k=N/2)

D. Mel Frequency Wrapping (MFW)

The Mel Frequency Warping process is a process of wrapping the signal spectrum using a triangular filter bank. Because the sound signal is different from human auditory perception, where the sound signal does not have a frequency with a linear scale. Therefore, adjustments are needed to the human auditory perception which is linear in the

feature extraction process to improve recognition performance. As a reference, the scaling between frequency in Hz and the mel scale is linear at frequencies below 1000 Hz and logarithmic at frequencies above it. To change the sound frequency to mel frequency, the following equation is used [12].

Input Gate formulation:

$$Y[i] = \sum_{j=1}^{G} T[j] H_i[j] \tag{5}$$

Table 5. The Mathematical Notation of Input Gate

Notation	Description
$Y[i]$	The calculation result of the i-th frequency wrapping mel
G	The sum of the spectrum magnitudes (G N)
$T[j]$	FFT result
$Hi[j]$	filterbank coefficient at frequency j (1 ≤ i ≤ E)
AND	Number of channels in the filterbank

E. Discrete Cosine Transform (DCT)

DCT is the last step of the main process of MFCC feature extraction. The basic concept of DCT is to decorrelate the mel spectrum so that it produces a good representation of the local spectral properties. The concept of DCT is the same as the inverse fourier transform. However, the results of DCT are close to PCA (Principle Component Analysis). PCA is a classical statistical method that is widely used in data analysis and compression. This is what often causes DCT to replace the inverse fourier transform in the MFCC Feature Extraction process [15]. DCT is assumed to replace the inverse fourier transform in the MFCC feature extraction process. The results of this DCT are the features needed by the author to carry out the analysis process for voice recognition. DCT uses Equation 6 [17].

$$C_m = \sum_{k=1}^{K} (\log_{10} Y[k] \cos\left[m(k - \frac{1}{2}) \frac{\pi}{K} \right]; m = 1,2,\ldots,K. \tag{6}$$

Notation	Description
S_K	the output of the filterbank process at index k
K	expected number of coefficients

F. Cepstrum

Cepstrum is the opposite term for spectrum. Cepstrum is commonly used to obtain information from a human spoken voice signal. In this final step, the log mel spectrum is converted to cepstrum using Discrete Cosine Transform (DCT). The result of this process is called MFCC 7. The result of the DCT function is cepstrum which is the final result of the feature extraction process. However, to improve the quality of recognition, the cepstrum resulting from DCT must undergo cepstral lifting [17].

$$w[n] = \left\{1 + \frac{L}{2} \sin\left(\frac{n\pi}{L}\right)\right\} \tag{7}$$

Notation	Description
L	the sum of cepstral coefficients
N	index of cepstral coefficients

3.2 HMM

HMM is a dual stochastic process with an underlying stochastic process that is unobservable (hidden), but can only be observed through a series of other stochastic processes that produce the sequence of symbols presented [18]. HMM is a stochastic model that describes two relationships between variables, namely unobserved variables (hidden state from time to time, and observed variables (observable state) [19]. HMM is suitable for the classification of one- or two-dimensional signals and can be used when information is incomplete or uncertain. To use HMM, we need a training phase and a testing phase. For the training phase, we usually work with the Baum-Welch algorithm to estimate the 8 parameters. In general, HMM consists of the following elements:

1. Set of observation output values$O = \{THE_1 , THE_2 , \ldots, THE_M\}$, Where $M$is the number of observation symbols.
2. Collection of states $= \{1,2, \ldots, N\}$. Where N states the number of states available in the HMM.
3. The set of transition probabilities between states. It is assumed that the next state depends on the current state. This assumption makes the calculation process easier and more efficient to perform. Transition probabilities can be expressed by a matrix$A = \{a_{ij}\}$, where a_ij is the transaction probability of state$i$the state of$j$. For example,$a_{ij} = P(S_{t-1} - 1 = i), 1 \leq i, j \leq N$ Where $St$is the state at time t-t.
4. The set of external probabilities $B = \{bi(k)\}$at each state. Also called the emission probability, $bi(k)$ is the probability of the output symbolokon state i which is defined as$b_i\ (k) = P(S_t = i)$Where $vt$is a symbol of observation at the time of t.
5. Initial state collection $\pi = \{\pi i\}$, Where $\pi i$is the probability of state$i$become the initial state in the HMM state sequence. Probabilistic parameters in Hidden Markov [20].

The type of HMM used in this study is a left-to-right type HMM with the number of states tested being 2 to 5 states. Figure 2.2 shows the HMM model of a system built with 4 states [16].

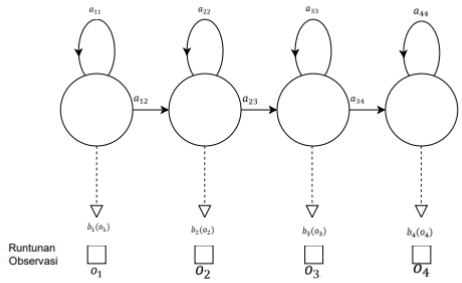

Figure 2.5 Left-right type HMM model with 4 states

The parameters used to build an HMM model are as follows:

$$\lambda = A, p, m \quad (8)$$

Notation	Description
A	state transition probability matrix
	initial state probability distribution
	average of a series of observations

The Baum-Welch algorithm is very supportive in the training stage using multiple observation sequences so it is very suitable for the problems faced [21]. The steps of the Baum-Welch algorithm are;

a. Forward procedure

In the forward procedure it can be defined as:

$$\alpha_t(i) = P(THE_1, THE_2, \ldots THE_t, i_t = i \mid l)$$
$$\alpha_t(i) \text{ can be calculated as follows}$$
$$\alpha_t(i) = \pi_i b_i(THE_t)$$
$$a_{t+1}(i) = b_j(THE_{t+1}) \sum \sum_{i=1}^{N} a_t(i).a_{ij} \quad (9)$$

b. Backward procedure

In the backward procedure it can be defined as;

$$\beta_t(i) = P(THE_{t+1}, THE_{t+2}, \ldots THE_T, i_t = i \mid l)$$
$$\beta_T(i) = 1, 1 \le i \le N$$
$$\beta_{t+1}(i) = b_j(THE_{t+1}) \sum \sum_{i=1}^{N} \beta_t(i).a_{ij} \quad (10)$$

3.3 Viterbi Algorithm

The steps in the Viterbi algorithm to find the best state sequence are as follows;

1. Initialization

$$\delta_1(i) = \pi_1 B_1(THE_1), 1 \le i \le N$$
$$\psi_1 = 0 \quad (11)$$

(2.6)

2. Rectancy

$$\delta_1(i) = a_{ij}]^* b_j(THE_t)$$
$$\psi_t(i) = a_{ij}]$$
For
$$2 \le t \le T$$
$$1 \le j \le N \quad (12)$$

3. Termination

$$P(\lambda) =]$$
$$q_t{}^* =] \quad (13)$$

It should be noted that the Viterbi algorithm is similar (except for the backtracking step) in its implementation to the computation. The main difference is that the maximization process in equation 12 over the previous state is used instead of the summation procedure. It should also be clear that the lattice (or trellis) structure efficiently implements the computation of the Viterbi procedure [18].

In this study, we conduct sound signal reading using the audio read command in the Matlab software. At this stage all the required voice data will be read at once, namely all training data and test data. All sound samples in *.wav format will be converted into Time Domain and Frequency Domain. Taking statistical values in the time domain with sound data is done to find out the sound data that will characterize the data. This is done in order to distinguish each data, which will be used in the classification process with the Mel

Frequency Cepstral Coefficients method. The values taken in the time domain are the average value and the standard deviation value of the voice data.

In this research, the MFCC method algorithm is used, which is one of the feature extraction methods used in the field of sound processing. This method is used for a process that converts sound signals into several parameters [14]. (MFCC) to convert time-domain signals to frequency-domain signals. To find out the characteristics of voice data, a voice signal is needed in frequency form and in spectral form and the magnitude value of each frequency of the voice emotion signal. The spectral data obtained from the MFCC results is then calculated for its statistical value. This is done to get the characteristics of the data to be used as an attribute for classifying data using the HMM method. This parameter will be used for HMM grouping.

4. Experimental Setup

4.1. Speaker Introduction

In the preparation of this research, the system built is a speaker recognition system where which system aims to recognize the identity of a speaker from the sound signal entered into the system. In the first stage, MFCC feature extraction is carried out, where in this MFCC feature extraction there are several stages, namely each sound signal is recorded with a specified time duration, after which framing preprocessing is carried out with overlapping. So that each word frame is obtained. Furthermore, the calculation of MFCC coefficients for each frame is carried out. So that each sound will be converted into a series of observations. Furthermore, the results of sound feature extraction act as input to the HMM system, referred to as an observation sequence (O). Each observation sequence is then determined by the observation probability value bj (Ot). The result is an HMM model for each speaker HMM model (λ). After the HMM model is obtained, the probability of the observation sequence model against the speaker HMM model (λ) is calculated with the Viterbi algorithm. The result of this model probability calculation is used as the similarity value.

4.2. Data Collection

The data collection process is carried out in a conducive recording environment, quiet and free from external noise to ensure optimal recording quality. The data used in the study are voice recordings from 30 speakers, each speaker says the word "PRESENT" 50 times so that 1500 data will be obtained, where the first 20 data of each speaker are used as test data and the last 30 data of each speaker are used as training data. The recording process is carried out using a smartphone with a distance of 10 cm - 15 cm from the sound source. The voice data from the recording is used for the feature extraction process using the MFCC method. The results of MFCC feature extraction will be used as an input dataset in the HMM method used for classification. The data that has been obtained is processed to determine the acoustic characteristics of the sound signal, determine the unique sound pattern of each speaker, and determine the accuracy of the system's success in recognizing the correct speaker.

4.3. Pre-Processing

In this pre-processing stage, the voice recording data will be reduced to improve the quality of the voice data, and prepare the data in the right form for the feature extraction process. At the initial stage, the voice data will be selected based on the amount of noise value in the voice recording. At this stage, the existing data will be converted into a .wav extension using the audacity sound processing program. Then the data will be cut to an

equal duration of 10 seconds. Performed in calm conditions and signals in a straight line and close to zero.

4.4. Feature Extraction

Voice data feature extraction is an advanced stage to process raw voice recording data in the .wav extension. Sound data feature extraction is performed using the MFCC method using MATLAB software. The flow of data feature extraction with 4 main stages, namely (1) reading all wav files in one folder, (2) retrieving the statistical value of sound data in the time domain, (3) transforming wav files into frequency domain form using the Mel Frequency Cepstral Coefficients method, (4) retrieving the statistical data value of sound data needed in the frequency domain.

5. Result and Analysis

5.1. Discussion

At this stage, the results of testing the training and test data as a whole from state 1 to state 5 are analyzed. Where the training data is done on the first 20 voice data of each speaker and the test data is done on the last 10 voice data of each speaker. After the implementation is carried out, an analysis of the experimental results is carried out which is used to evaluate the model. This evaluation aims to measure how well the model classifies an object. After HMM training on training data, the best weight vector is obtained which is then used for testing in recognizing voice data patterns. Thus, the class of the test data can be determined. To test the accuracy of the model, the training data is processed using the Baum-Welch algorithm, while the test data is evaluated using the Viterbi algorithm to measure the accuracy of the model. The results of this test yielded the best value of the overall evaluation.

5.2. State Recognition System Testing 1

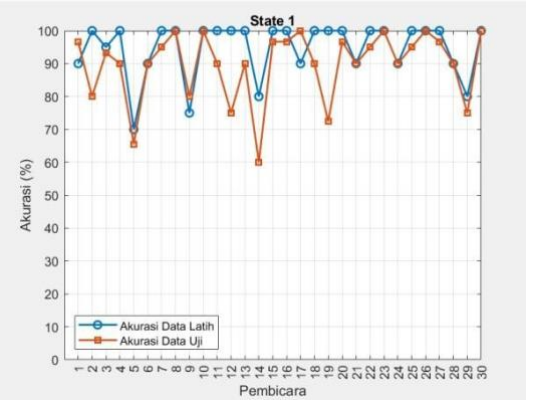

Fig. 5.1. Graph of testing accuracy results of training data and state 1 test data

Fig 5.1 shows the results of overall data testing in state 1. The accuracy of the average value of the training data is 89.50%, while the accuracy of the average value of the test data is 83.63%, with a difference of about 5.87% between these two values indicating that the model tends to be better at recognizing patterns on familiar data in the training data, but experiences a slight decrease in performance when tested on new data in the test data.

5.3. State Recognition System Testing 2

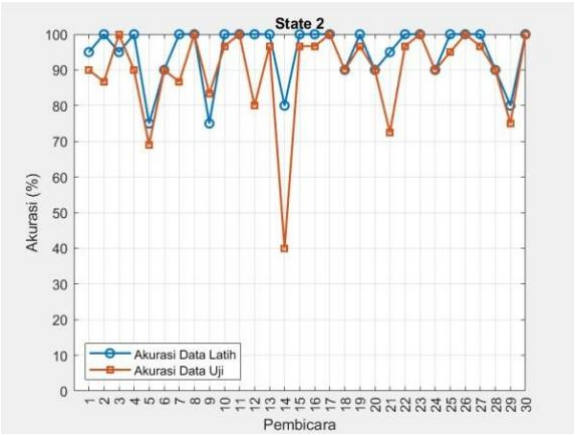

Fig. 5.2. Graph of testing accuracy results of training data and state 2 test data

Fig 5.2. shows the results of overall data testing in state 2. The average accuracy of the training data is 95.00%, while the average accuracy of the test data is 86.77%, with a difference of about 8.23%. Between these two values, it shows that the model tends to be better at recognizing patterns on familiar data in the training data, but experiences a slight decrease in performance when tested on new data in the new data.

The results of testing data in state 2 tend to be better than in state 1, although the difference between test data and training data in state 2 tends to be more than in state 1. The overall accuracy of the training data reaches 95.00%, which indicates that the model is able to recognize patterns from the data used in the training process very well. Meanwhile, the overall accuracy of the test data is 86.77%, which is lower than the training data, with a difference of about 8.23%. This difference indicates a symptom of overfitting, where the model overfits itself to the training data, but is less able to generalize to new data.

5.4. State Recognition System Testing 3

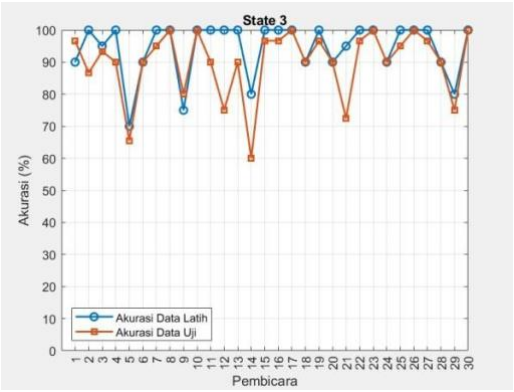

Fig. 5.3. Graph of testing accuracy results of training data and state 3 test data

Fig 5.3 shows the results of overall data testing in state 3. The average accuracy of the training data is 95.17%, while the average accuracy of the test data is 89.46%, with a difference of about 8.23%. Between these two values, it shows that the model tends to be

better at recognizing patterns on familiar data in the training data, but experiences a slight decrease in performance when tested on new data in the new data.

The results of data testing in state 3 show a more stable and high test accuracy than in states 1 and 2. This shows that the model's ability to generalize is better and the balance between training data and test data even though there is still a decrease in some speakers and in this state is suitable for variable data models.

5.5. State Recognition System Testing 4

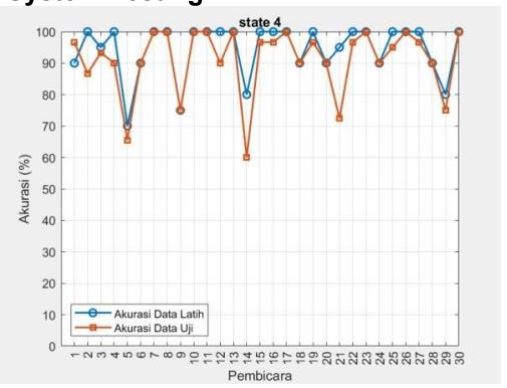

Fig. 5.4. Graph of testing accuracy results of training data and state 4 test data

Fig 5.4 shows the results of overall data testing in state 4. The average accuracy of the training data is 95.67%, while the average accuracy of the test data is 88.79%, with a difference of about 8.23%. Between these two values, it shows that the model tends to be better at recognizing patterns on familiar data in the training data, but experiences a slight decrease in performance when tested on new data in the new data.

The results of testing data in state 4 show similar characteristics to State 3, with almost perfect training accuracy and fairly high test accuracy. Some speakers such as the 5th and 14th still experienced a decline, but not as severe as in State 2. The model in this state shows a good balance, but there is still room for improvement, especially on speakers whose performance is unstable. This may indicate differences in voice characteristics that are not optimally captured by the model.

5.6. State Recognition System Testing 5

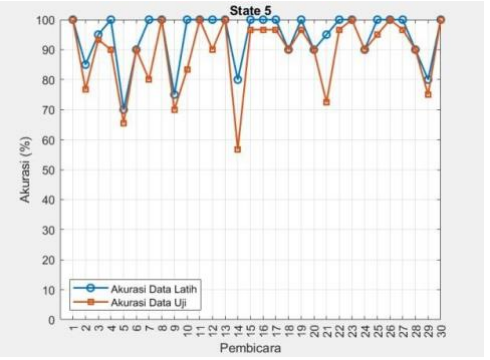

Fig. 5.5. Graph of testing accuracy results of training data and state 5 test data

Fig 5.5 shows the results of overall data testing in state 5. The average accuracy of the training data is 94.50%, while the average accuracy of the test data is 88.79%, with a difference of about 8.23%. Between these two values, it shows that the model tends to be better at recognizing patterns on familiar data in the training data, but experiences a slight

decrease in performance when tested on new data in the new data.

The results of testing the data in state 5 provide the most stable accuracy results overall, both on the training and test data. The decrease in test accuracy is still visible in some speakers (such as the 13th speaker), but in general the variation is smaller. This shows that the model configuration in State 5 performs well in terms of both training and testing. The model in this state can be considered as the best candidate among all states based on the balance between training and test accuracy.

5.7. State Recognition System Testing

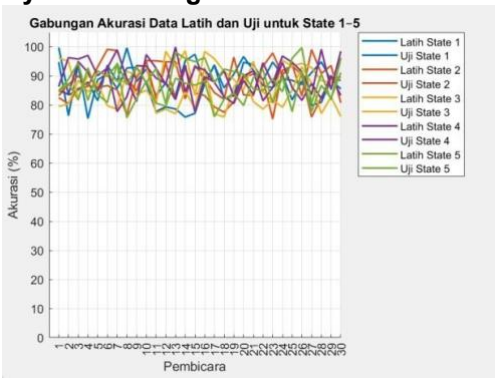

Fig. 5.6. Graph of testing accuracy results of training data and state 1-5 test data

The overall test results showing the accuracy of the training and test data for each state (1-5) show a general pattern that the accuracy of the training data tends to be high and stable across speakers and states, while the accuracy of the test data is more variable and fluctuates depending on the state and speaker. The overlapping line pattern for the training accuracy indicates the consistency of model learning, but the sizable difference between the test and training curves at some points indicates potential overfitting. The lines for test accuracy in some states show a sharp drop, indicating that the generalization performance of the model to new data is not optimal in all state configurations. However, some states such as State 3 and 5 appear to show a more even distribution of higher accuracy on the test data, indicating that the model configurations have better performance in general.

6. Conclusion

According to the experimental result, the average accuracy of the model can achieve excellent training performance. The average training accuracy exceeds 90%, indicating the model's strong capacity to learn and fit the training data effectively. This model performance shows robust learning across different datasets. However, the model's accuracy on test data exhibits greater variability, with average values ranging from 86% to 89%. This fluctuation indicates that, despite effective learning from training data, the model’s generalization ability to unseen data remains somewhat unstable. The performance gap between training and test accuracy highlights the need for further optimization to enhance the model’s robustness and generalization.

References

[1] S. Hidayat, M. Tajuddin, S. A. Alodiayusuf, J. Qudsi, and N. N. Jaya, "Wavelet detail coefficient as a novel Wavelet-MFCC feature in text-dependent speaker recognition system," *IIUM Eng. J.*, vol. 23, no. 1, pp. 68–81, 2022, doi: 10.31436/iiumej.v23i1.1760.

[2] B. Triandi, H. Mawengkang, and S. Efendi, "Comparison of speaker voice feature extraction techniques," *J. RESTI*, vol. 12, pp. 33–42, 2021.

[3] B. S. El and T. Pangaribowo, "Speech recognition application for computer software security using MFCC (Mel Frequency Cepstrum Coefficients) and HMM (Hidden Markov Model) methods," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 11, no. 1, pp. 10–18, 2020.

[4] G. Ajinurseto, L. O. Bakrim, and N. Islamuddin, "Application of Mel Frequency Cepstral Coefficients method in desktop-based speech recognition system," *Infomatek*, vol. 25, no. 1, pp. 11–20, Jun. 2023, doi: 10.23969/infomatek.v25i1.6109.

[5] Y. R. Prayogi, "Modification of the MFCC method for speaker identification in noisy environments," *Jointecs*, vol. 4, no. 1, pp. 13–21, 2019, doi: 10.31328/jointecs.v4i1.999.

[6] F. Fadlisyah, S. Safwandi, and M. A. Altharizka, "Qur'anic verse recognition system on Surah Al-Qari'ah using Hidden Markov Model (HMM) method," *Techsi*, vol. 12, no. 1, pp. 96–103, 2020, doi: 10.29103/techsi.v12i1.2151.

[7] D. Jollyta, D. Oktarina, and J. Johan, "Case review of speech recognition model: Hidden Markov Model," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 2, pp. 202–210, Aug. 2020, doi: 10.26418/jp.v6i2.39231.

[8] S. Hidayat, A. S. Anas, S. Agrippina, A. Yusuf, and M. Tajuddin, "Speaker recognition system with Wavelet-MFCC method and Hidden Markov Models (HMM) classifier," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 1, pp. 119–126, 2021, doi: 10.25126/jtiik.202183284.

[9] S. Permana, Y. I. Nurhasanah, and A. Zulkarnain, "Implementation of MFCC and DTW methods for recognizing male and female voice types," *Mind J.*, vol. 3, no. 1, pp. 49–63, 2018.

[10] Y. I. Nurhasanah, Andriana, and D. Permatasari, "Speaker recognition to determine gender using the MFCC and VQ methods," *Mind J.*, vol. 1, pp. 34–47, 2017, doi: 10.26760/mindjournal.v1i1.22.

[11] Q. Nada, C. Ridhuandi, P. Santoso, and D. Apriyanto, "Speech recognition with Hidden Markov Model for Hijaiyah letter recognition and pronunciation," in *Proc. Nat. Semin. Informatika*, 2019.

[12] S. Hidayat, R. Hidayat, and T. B. Adji, "Indonesian speech recognition system based on syllables using MFCC, Wavelet and HMM," in *Proc. Int. Semin. Intell. Technol. Appl. (ISITIA)*, 2015.

[13] M. N. Rabbani, A. Rizal, I. Fiky, and Y. Suratman, "Implementation of voice recognition-based key using Mel Frequency Cepstral Coefficient (MFCC) method," in *Proc. Nat. Semin. Komput. dan Informatika*, 2016.

[14] Heriyanto, S. Hartati, and A. E. Putra, "Feature extraction Mel Frequency Cepstral Coefficient (MFCC) and mean coefficient for checking Al Quran reading," *J. Teknol. Inf.*, vol. 15, no. 2, pp. 85–92, 2018.

[15] T. Nasution, "Mel Frequency Cepstrum Coefficients (MFCC) method for recognizing speech in Indonesian," *J. Teknol. dan Sistem Komput.*, vol. 1, no. 1, pp. 9–13, 2012.

[16] B. Darmawan and S. Ariessaputra, "HMM speaker recognition and verification system," *J. Inform. dan Sistem Inf.*, vol. 4, no. 2, pp. 45–52, 2018.

[17] D. Andriana, "Software to open applications on computers with voice commands using the Mel Frequency Cepstrum Coefficients method," *J. Ilm. Komput. dan Informatika (KOMPUTA)*, vol. 21, no. 1, pp. 10–16, 2013.

[18] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986, doi: 10.1109/MASSP.1986.1165342.

[19] [19] M. Gultom, D. Alamsyah, S. G. MDP, and J. Rajawali, "Application of Hidden Markov Model (HMM) in speaker recognition," in *Proc. Semin. Nas. Teknol. Inform. dan Komunikasi (SENATIK)*, 2014.

[20] M. Susant, B. Susilo, and D. Andreswari, "Speech-to-text application using Mel Frequency Cepstral Coefficient (MFCC) and Hidden Markov Model (HMM) methods in searching ICD-10 codes," *J. Rekursif*, vol. 6, no. 2, pp. 92–98, 2018.

[21] B. Robbiyanto and R. Magdalena, "Design and analysis of speech processing system for a deaf person using Hidden Markov Model method and Mel-Frequency Cepstral Coefficient," in *Proc. Semin. Nas. Sains dan Teknol. Inform. (SENSASI)*, 2019