EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 2 (2024) 51-60
https://publisher.uthm.edu.my/periodicals/index.php/eeee

A MATLAB-Based Time Delay Neural Network (TDNN) Approach to Speaker Recognition for Voice Biometric Attendance Systems: Model Development and GUI Design

Chin Chieh Vee[1], Zarina Tukiran[1]*, Rafizah Mohd Hanifa[2]

[1] *Internet of Things Focus Group, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA*

[2] *Center of Diploma Studies, Universiti Tun Hussein Onn Malaysia Pagoh Campus, Pagoh, 84600, MALAYSIA*

*Corresponding Author: zarin@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.02.007

Article Info

Received: 27 June 2024
Accepted: 25 September 2024
Available online: 30 October 2024

Keywords

Speaker recognition, Time Delay Neural Network (TDNN), MATLAB GUI, Attendance System

Abstract

Traditional attendance systems often suffer from inefficiency, inaccuracy, and vulnerability to fraud. Utilization of voice biometrics can help to mitigate these downsides. This paper presents the development and design of a MATLAB-based voice biometric attendance system that is inspired by MathWorks' x-vector implementation. The system utilizes a Time Delay Neural Network (TDNN) for speaker recognition which has been trained on a 100 hour subset of the LibriSpeech dataset, with Mel-Frequency Cepstral Coefficients (MFCCs) being used as input features. A graphical user interface (GUI) has been developed within MATLAB App Designer which provides features such as user registration, login, voice enrollment, and attendance recording. The system also incorporates a local database for user enrollment data and employs text-dependent voice enrollment for enhanced user experience and accuracy. This work demonstrates the successful integration of the x-vector approach into a functional attendance system, highlighting its practicality and feasibility for real-world applications.

1. Introduction

Voice biometrics has proven to be a viable method for identity verification, appreciated for its universality and the minimal intrusion it imposes on users (Hanifa et al., 2021). As technological integration into daily operations becomes more pronounced, the demand for efficient authentication solutions grows, with voice biometrics offering a suitable pathway to meet these needs (Reynolds et al., 1995; Kinnunen et al., 2010). The evolution of voice biometric models, from foundational works to contemporary approaches like the TDNN and X-vector models (Waibel et al., 1989; Snyder et al., 2018), signifies a continuous refinement in speaker recognition techniques.

This paper focuses on the practical development of a MATLAB-based voice biometric attendance system, utilizing the TDNN architecture as its core, inspired by MathWorks' speaker recognition implementation (The MathWorks Inc., 2024). The system leverages the TDNN's ability to extract speaker-specific features, known as x-vectors, which exhibit improved stability under varying environmental conditions (Snyder et al., 2018). This approach, while not claiming to be the pinnacle of innovation, represents a solid advancement in the realm of speaker recognition for attendance systems.

Central to this system is the creation of a graphical user interface (GUI) within MATLAB, designed to pragmatically support essential functions such as user registration, login, voice enrollment, and attendance marking (Kishore et al., 2023; Masykuroh et al., 2021). The GUI, developed through MATLAB App Designer, serves as a practical implementation tool rather than a showcase of user-friendliness. Its purpose is to facilitate the operational aspects of the voice biometric attendance system, aligning with the system's overall goal of providing a reliable and efficient authentication mechanism (Amri et al., 2017; Bekkanti et al., 2021).

In the following sections, into the technical aspects of the TDNN-based speaker recognition system and the GUI's practical design. The paper explores the system's functionality and its potential impact on attendance management, contributing to the ongoing discourse on the practical applications of voice biometrics in authentication systems.

2. Methodology

The block diagram as shown in Fig. 1 illustrates the process of the development of the speaker recognition model.

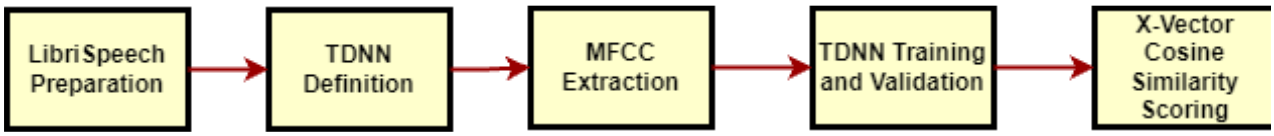


Fig. 1 *Development of the speaker recognition model*

2.1 LibriSpeech Preparation

In developing our speaker recognition model, we utilized the LibriSpeech dataset, focusing on the 100-hour clean subset for training, validation, and testing (Panayatov et al., 2015). This subset was prepared by dividing the audio into training, validation, and test sets using MATLAB's audioDatastore objects.

2.2 TDNN Definition

Our TDNN architecture, as illustrated in Fig 2, consists of five frame-level layers, each with 512 filters, followed by ReLU activation functions. This architecture effectively captures temporal dependencies in the input feature vectors, encoding both short-term and long-term speaker characteristics. The frame-level outputs are aggregated using a statistics pooling layer to generate a fixed-dimensional x-vector embedding, which serves as a compact representation of the speaker's vocal identity.

Layer	Description	Layer Context	Total Context	Input-by-Output
1	**1-d convolutional** batch normalization ReLu activation	$[t-2, t+2]$	5	$(5 \times numFeatures) - by - numFilters$
2	**1-d convolutional** batch normalization ReLu activation	$\{t-2, t, t+2\}$	9	$(3 \times numFilters) - by - numFilters$
3	**1-d convolutional** batch normalization ReLu activation	$\{t-3, t, t+3\}$	15	$(3 \times numFilters) - by - numFilters$
4	**1-d convolutional** batch normalization ReLu activation	$\{t\}$	15	$numFilters - by - numFilters$
5	**1-d convolutional** batch normalization ReLu activation	$\{t\}$	15	$numFilters - by - 1500$
6	**statistics pooling**	$[0, T)$	T	$(1500 \times T) - by - 3000$
7	**fully-connected** batch normalization ReLu activation	$\{0\}$	T	$3000 - by - numFilters$
8	**fully-connected** batch normalization ReLu activation	$\{0\}$	T	$numFilters - by - numFilters$
9	**fully-connected** softmax	$\{0\}$	T	$numFilters - by - N$

Fig. 2 *Definition of the TDNN. Adapted from Speaker Recognition Using x-vectors, by The MathWorks, Inc., 2024, https://www.mathworks.com/help/audio/ug/speaker-recognition-using-x-vectors.html*

2.3 MFCC Extraction

Evolution in Electrical and Electronic Engineering Vol. 5 No. 2 (2024) p. 51-60

MFCCs were extracted from the audio data, employing Hann windows to each frame to reduce spectral leakage, and the audioFeatureExtractor object from MATLAB using a similar method as seen in Rabiner and Schafer's (2007) *Introduction to Digital Speech Processing.* For this implementation, 30 MFCCs were extracted from audio files using 30 ms Hann windows with a 10ms hop. After the features were extracted, they were used to create transform audio datastores for training and evaluation, each prefixed with ads denoting audio datastore. These datastores include adsTrain for training audio datastore, adsValidation for validation audio datastore, adsTest for testing audio datastore, adsEnroll for enrolment audio datastore and adsDET for DET curve audio datastore.

2.4 TDNN Training and Validation

The TDNN model was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 256. Training involved a maximum of 5 epochs with early stopping based on the validation loss to prevent overfitting. Categorical cross-entropy loss function was used as the optimization objective. cosine similarity scoring was employed for authentication, using a 150-dimensional LDA projection matrix to enhance discriminative power between speaker classes.

2.5 X-Vector Cosine Similarity Scoring

The performance of the TDNN model was then evaluated using a Detection Error Tradeoff (DET) curve, with the average cosine similarity scores serving as the final decision metric and a predetermined threshold dictating acceptance or rejection (Hemmat, 2024).

2.6 Attendance System Design and Integration

The trained model was then loaded into the attendance system. Regarding the attendance system itself, Fig. 3 and Fig. 4 show the overall block diagrams of the system.

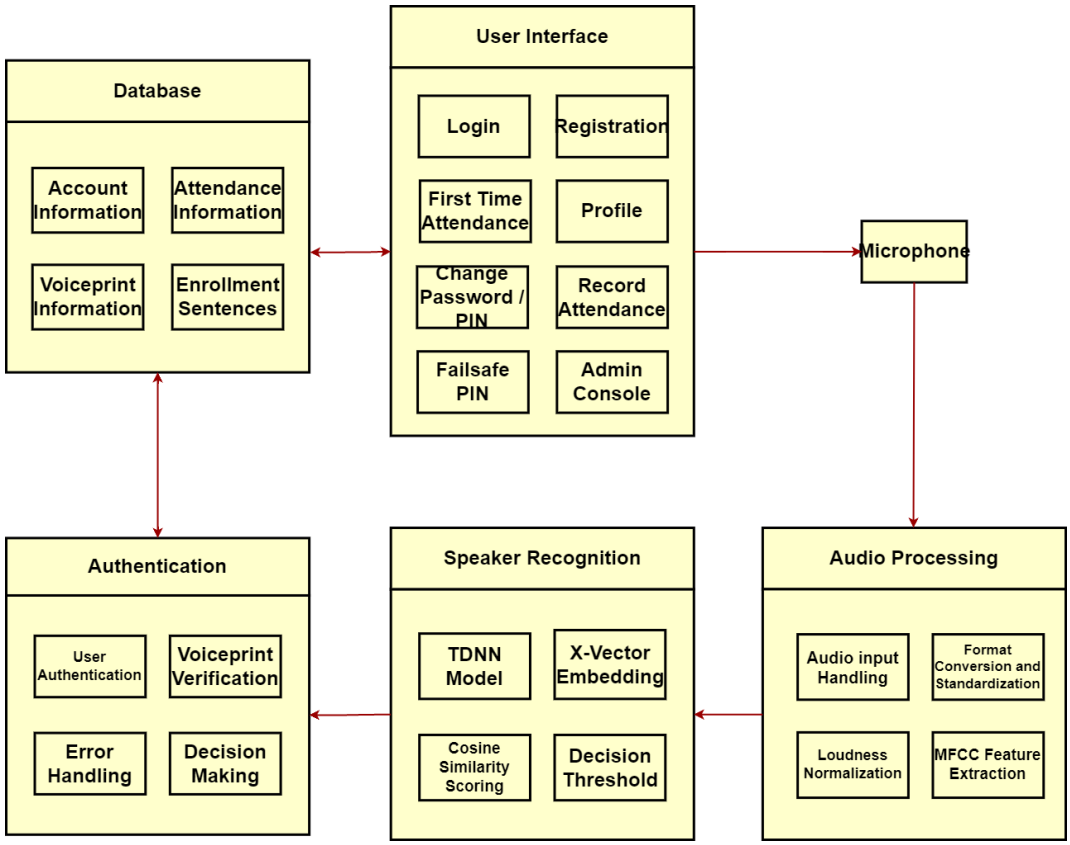


Fig. 3 *Attendance system block diagrams – the system architecture*

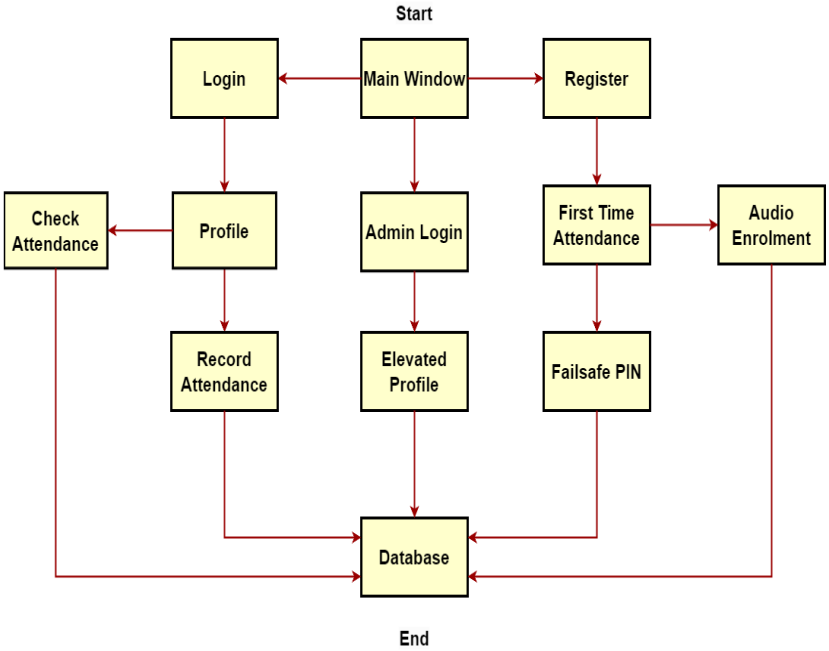


Fig. 4 *Visualisation of system connecting windows*

The biometric attendance system utilises a multiwindow GUI for user interaction wherein information is shared across windows, a local .xlsx database for storing user accounts, attendance records, and a core functionality set that includes account registration, audio recording for identity authentication, and attendance tracking. The attendance system itself is primarily software-based, with a physical microphone serving as the audio input device.

The system also incorporates text-dependent voice enrolment, where users are required during first time attendance to record three audio samples of the same sentence to create a voiceprint template. This approach, based on Microsoft's speaker recognition API recommendations (Microsoft Cognitive Services, n.d), ensures sufficient feature capture for accurate identification. Table 1 shows the enrolment sentences available to the user. Additionally, users must enter a failsafe PIN. Fig. 5 shows the flowchart for first time attendance.

Table 1 *Enrolment sentences*

Number	Sentence
1	The quick brown fox jumped over the sleepy dog.
2	Now is the time to relax, unwind and enjoy the day.
3	An apple a day keeps the doctors far away.
4	Love and peace bring joy and harmony to the world.
5	The tiny blue bird flew past the giant green tree.
6	Gentle waves dance along the sandy shorelines.
7	Silly jokes bring laughter to everyone in the room.
8	Happy thoughts create smiles all around the world.
9	The best kind of meal is the meal shared with friends.

Evolution in Electrical and Electronic Engineering Vol. 5 No. 2 (2024) p. 51-60

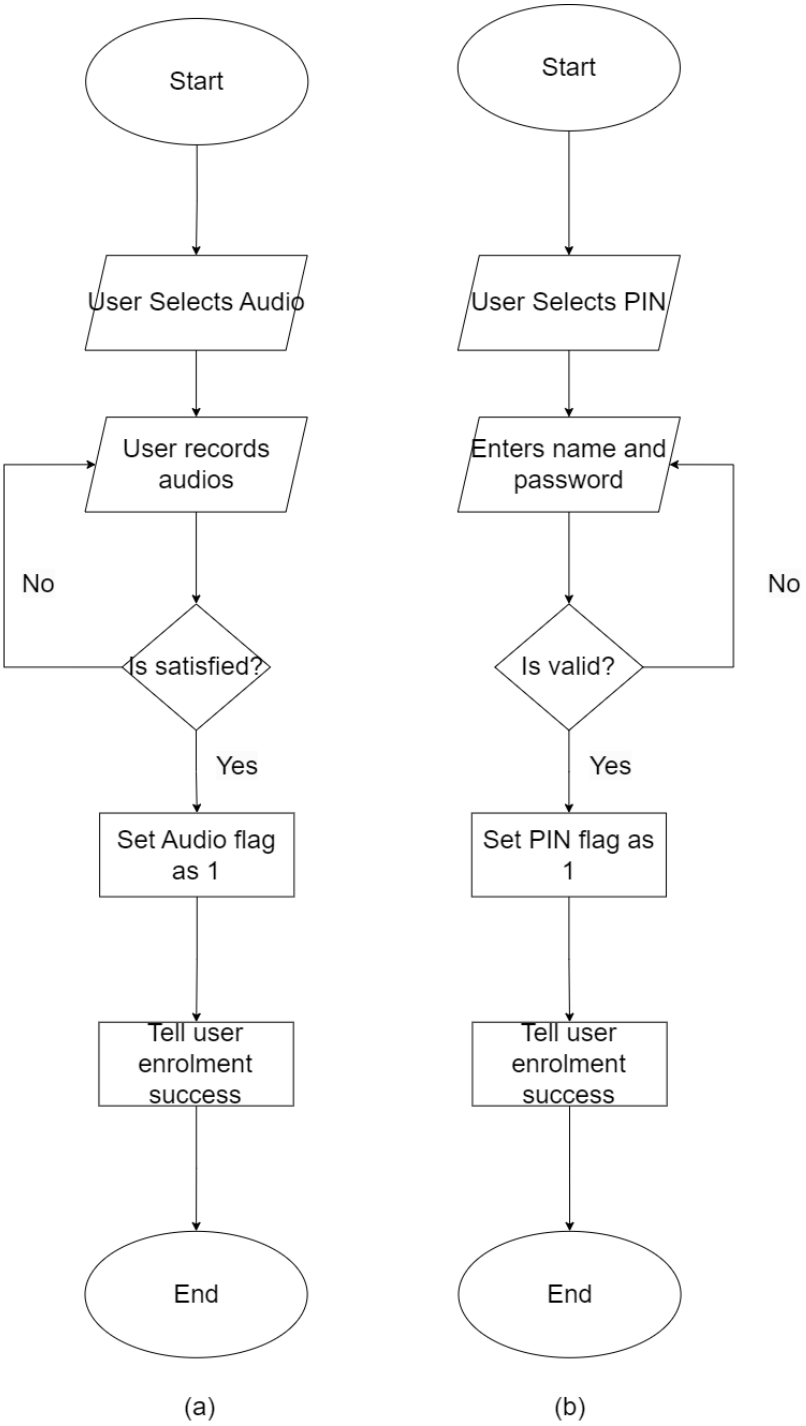


Fig. 5 *First time attendance flowchart for (a) Record button and (b) PIN button*

For administrators, a dedicated console provides access to account and attendance information, the enrolment table, and the ability to set attendance. This console allows administrators to edit database values in real-time and playback user audio files, offering a comprehensive toolset for managing the system.

Fig. 6 shows the attendance marking flowchart, wherein an obtained threshold from the DET curve is used to compare with the cosine similarity score. If the score is greater than the threshold, user's attendance is marked, but if it's less than the threshold, user is asked to enter failsafe PIN as a backup, and if failsafe PIN fails, then attendance is not taken.

Additionally, to assess the performance of the system, the execution times of key functions were measured using MATLAB's tic and toc commands. The functions that were identified for assessment were the account login, user enrolment and attendance marking functions.

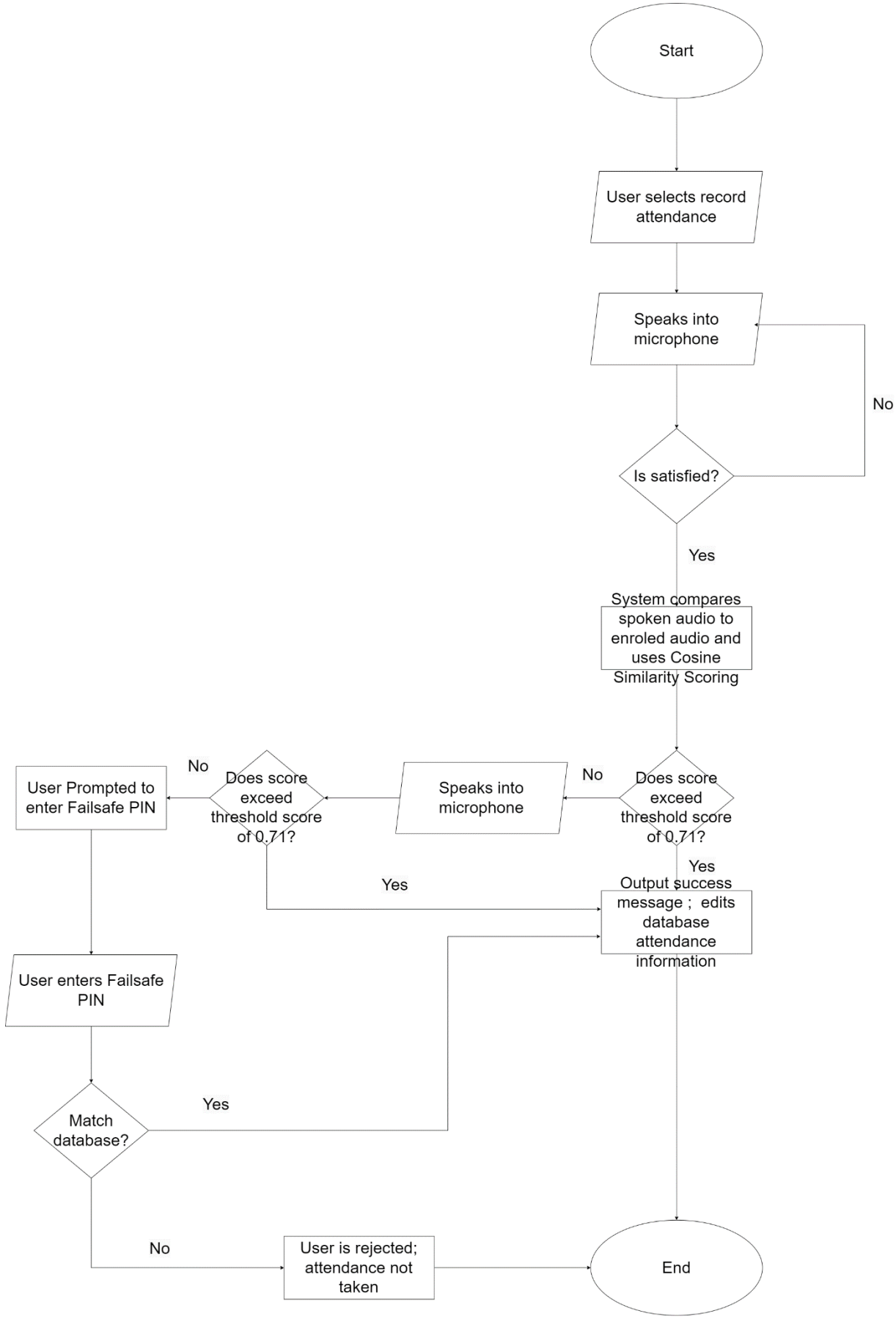
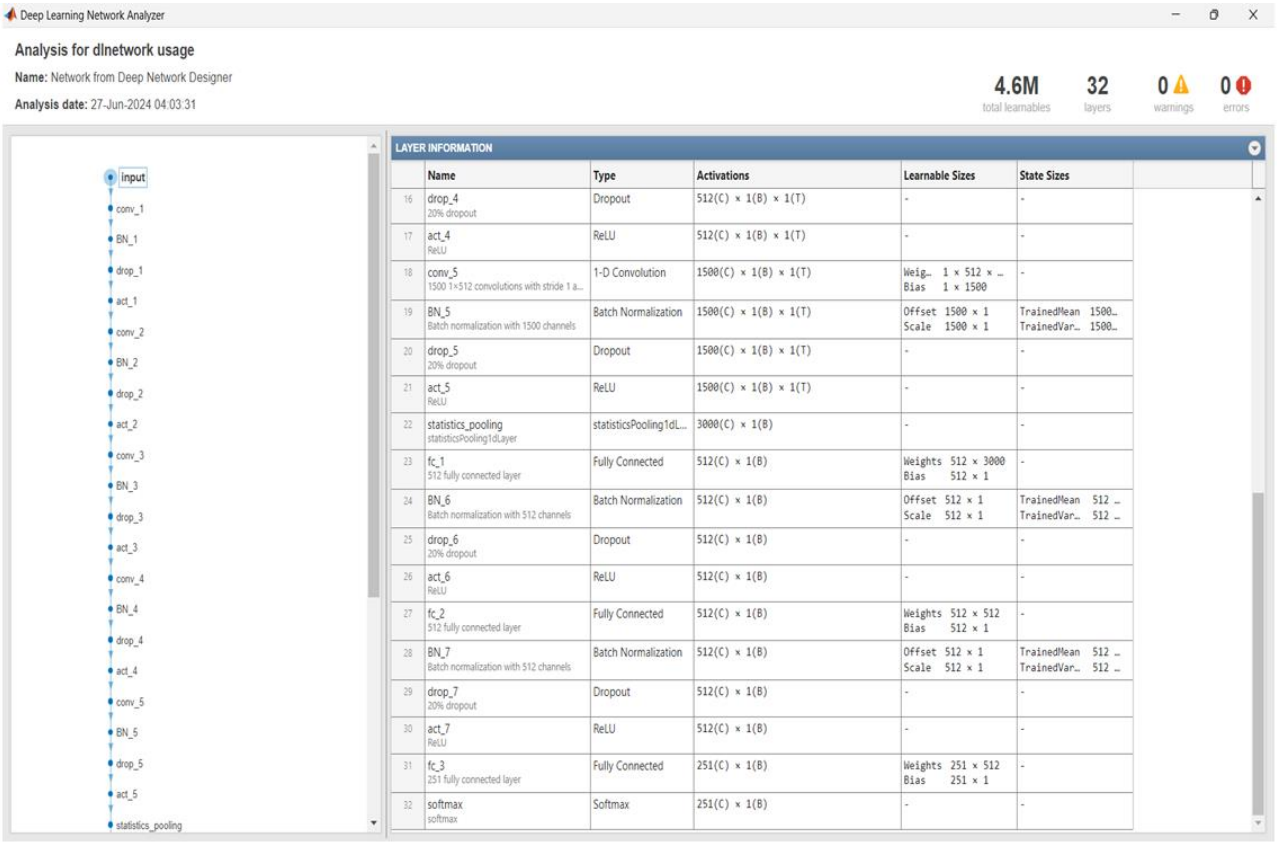


Fig. 6 *Attendance marking flowchart*

Evolution in Electrical and Electronic Engineering Vol. 5 No. 2 (2024) p. 51-60

3. Result and Discussion

Fig. 7 shows the result of the visualisation of the TDNN structure using MATLAB's Deep Network Designer's analyser tool after implementing it programmatically. The model contains 4.6 million learnables and 32 layers and occupies 18MB of space. Layer 23 "fc_1" is used for the x-vector embeddings.



LAYER INFORMATION

	Name	Type	Activations	Learnable Sizes	State Sizes
16	drop_4 20% dropout	Dropout	512(C) × 1(B) × 1(T)	-	-
17	act_4 ReLU	ReLU	512(C) × 1(B) × 1(T)	-	-
18	conv_5 1500 1×512 convolutions with stride 1 a...	1-D Convolution	1500(C) × 1(B) × 1(T)	Weig... 1 × 512 × ... Bias 1 × 1500	-
19	BN_5 Batch normalization with 1500 channels	Batch Normalization	1500(C) × 1(B) × 1(T)	Offset 1500 × 1 Scale 1500 × 1	TrainedMean 1500... TrainedVar... 1500...
20	drop_5 20% dropout	Dropout	1500(C) × 1(B) × 1(T)	-	-
21	act_5 ReLU	ReLU	1500(C) × 1(B) × 1(T)	-	-
22	statistics_pooling statisticsPooling1dLayer	statisticsPooling1dL...	3000(C) × 1(B)	-	-
23	fc_1 512 fully connected layer	Fully Connected	512(C) × 1(B)	Weights 512 × 3000 Bias 512 × 1	-
24	BN_6 Batch normalization with 512 channels	Batch Normalization	512(C) × 1(B)	Offset 512 × 1 Scale 512 × 1	TrainedMean 512 ... TrainedVar... 512 ...
25	drop_6 20% dropout	Dropout	512(C) × 1(B)	-	-
26	act_6 ReLU	ReLU	512(C) × 1(B)	-	-
27	fc_2 512 fully connected layer	Fully Connected	512(C) × 1(B)	Weights 512 × 512 Bias 512 × 1	-
28	BN_7 Batch normalization with 512 channels	Batch Normalization	512(C) × 1(B)	Offset 512 × 1 Scale 512 × 1	TrainedMean 512 ... TrainedVar... 512 ...
29	drop_7 20% dropout	Dropout	512(C) × 1(B)	-	-
30	act_7 ReLU	ReLU	512(C) × 1(B)	-	-
31	fc_3 251 fully connected layer	Fully Connected	251(C) × 1(B)	Weights 251 × 512 Bias 251 × 1	-
32	softmax softmax	Softmax	251(C) × 1(B)	-	-

Fig. 7 *Visualisation of TDNN architecture*

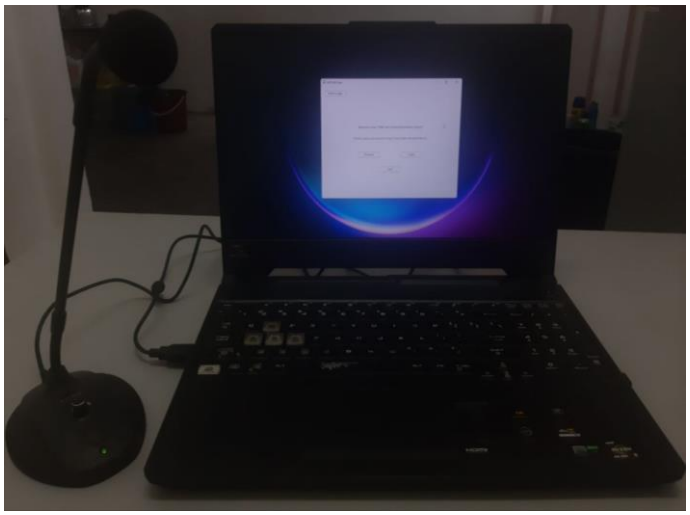
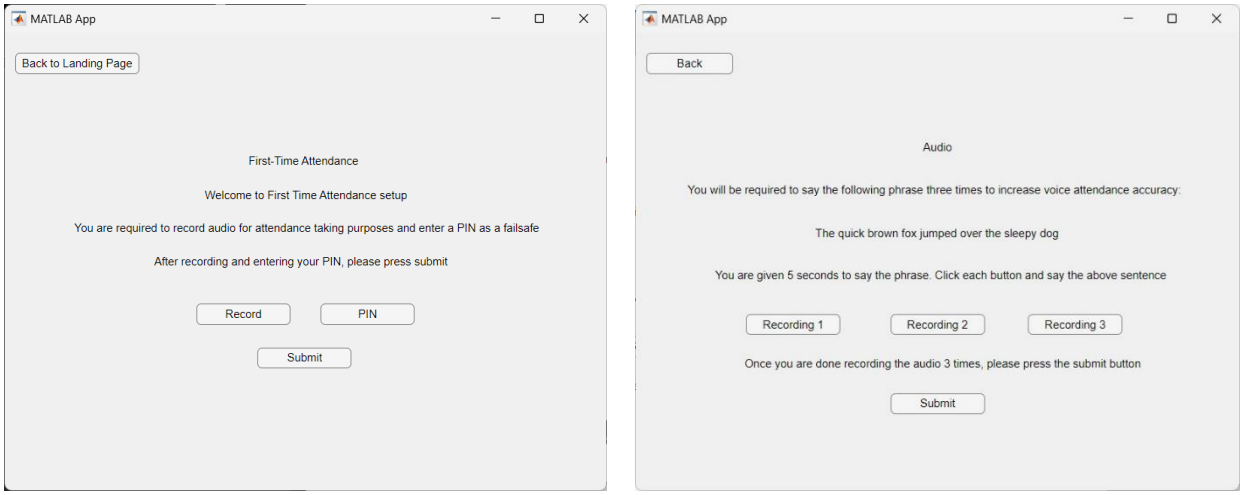


Fig. 8 *Attendance system setup with desktop microphone*

Fig. 8 shows the setup of the attendance system. Since the system is mostly software based, a standard desktop microphone was used for the audio input. In this case, a Fifine K052 microphone was used for the system.

Fig. 9 shows examples of the GUI windows of the system, mainly the first-time attendance functionality and the record audio functionality. Although the system is currently PC based only, the total space occupied by the system is around 25MB, which indicates the possibility of implementing this system in other platforms such as in the form of mobile applications or within embedded systems.



(a) **(b)**

Fig. 9 *GUI windows for (a) First Time Attendance and (b) Record Audio*

Fig. 10 and Fig. 11 show the account info and attendance info sheets of the .xlsx database. The system will keep track of the users' enrolled sentence as well as their set flags to ensure a smooth onboarding experience for the users during enrolment. Additionally, the attendance sheet will keep track of the total number of attendance as well as attendance percentage, allowing for potential attendance visualisation functionality integration in future versions.

UID	Name	Password	PIN	FirstTimeAttendanceDone	AudioCompletionFlag	PINCompletionFlag	Sentence
0	Alpha	Omega	1337	1	1	1	1

Account Info | Attendance Info | Sentences | Settings

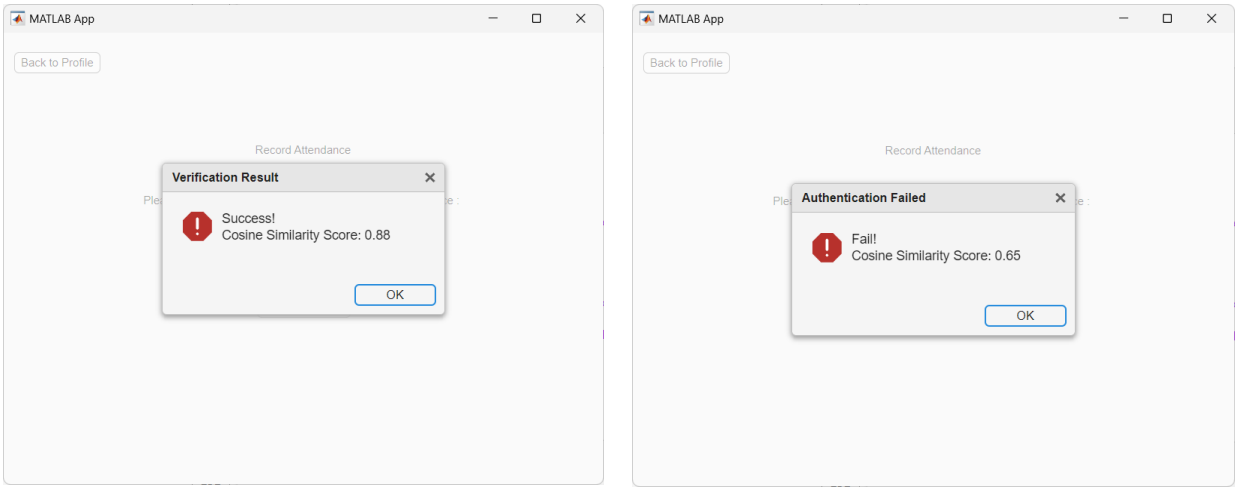
Fig. 10 *Account info database*

UID	Name	Total	% Attendance	28/05/2024	04/06/2024	11/06/2024	18/06/2024	25/06/2024	02/07/2024	09/07/2024	16/07/2024	23/07/2024	30/07/2024	06/08/2024	1
0	Alpha	19	86.36363636	P	A	A	P	A	P	P	P	P	P	P	P
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												
		0	#DIV/0!												

Fig. 11 *Attendance info database*

Fig. 12 shows the GUI windows for success and failure condition. After user signs into their account and records their sentence, the system compares the recorded audio to that of the enrolled X-vector voiceprint using cosine similarity scoring. If the recorded audio has a similarity score of more than the cutoff which was determined to be 0.7176 using DET, then the user's attendance is enrolled as indicated in Fig 10. Otherwise, the user is rejected.

Evolution in Electrical and Electronic Engineering Vol. 5 No. 2 (2024) p. 51-60



(a) **(b)**

Fig. 12 *GUI windows for (a) Success and (b) Failure*

Fig. 13 shows the Admin console window, with various buttons such as checking account info, attendance info, enrolment table information, editing user entries, setting attendance, select user audio files for playback and opening the .xlsx database.

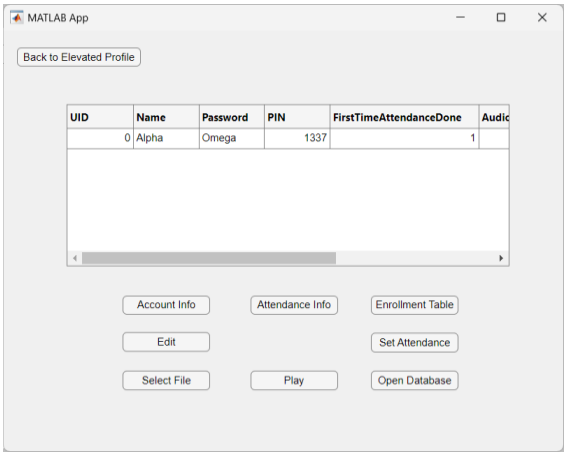


Fig. 13 *Admin console*

Table 2 shows the average time taken for the functions to be executed. Preliminary results indicate the system's potential for real-time use, with average processing times of 0.26 seconds for login, 0.45 seconds for enrolment, and 0.72 seconds for attendance marking (Table 2). While attendance marking may benefit from further optimization, the overall system demonstrates promising speed and efficiency.

Table 2 *Time taken for the functions*

Function	Average time taken (s)
Account login	0.263462
User enrolment	0.447517
Attendance marking	0.718547

4. Conclusion

In conclusion, this paper presents a prototype voice biometric attendance system that successfully integrates a TDNN model for speaker recognition with a user-friendly MATLAB App Designer GUI. Trained on a 100-hour subset of LibriSpeech, the system demonstrates the practical feasibility of the x-vector approach for real-world applications.

Preliminary results indicate promising functionality and speed, with average processing times under one second for key functions like login, enrollment, and attendance marking. The system's modest 25MB footprint

allows for flexible deployment, while the use of a standard desktop microphone underscores its software-centric design. A local .xlsx database effectively manages user data and attendance records, providing a foundation for future expansion with features such as attendance visualization.

While this work focuses primarily on system development and architecture, future work will prioritize a comprehensive quantitative evaluation of the system's performance under various real-world conditions. This evaluation will include testing the system's accuracy, robustness to noise and speaker variations, and real-time performance using a custom dataset of Malaysian speakers. This work successfully achieves its objective in laying the groundwork for voice-based attendance systems that harness the unique characteristics of human voices for identity verification.

Acknowledgement

The authors would like to express their sincerest gratitude to the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support throughout the completion of this work.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Chin Chieh Vee, Zarina Tukiran, Rafizah Mohd Hanifa;* ***data collection:*** *Chin Chieh Vee;* ***analysis and interpretation of results:*** *Chin Chieh Vee;* ***draft manuscript preparation:*** *Chin Chieh Vee, Zarina Tukiran, Rafizah Mohd Hanifa. All authors reviewed the results and approved the final version of the manuscript.*

References

Amri, U., Nik Hashim, N. N. W., & Hanif, N. (2017). Speech-based Class Attendance. IOP Conference Series: Materials Science and Engineering, 260(1), 012008. https://doi.org/10.1088/1757-899X/260/1/012008

Bekkanti, N., Busch, L., & Amman, S. (2021). Evaluation of Voice Biometrics for Identification and Authentication. SAE Technical Paper Series. https://doi.org/10.4271/2021-01-0262

Hanifa, R. M., Isa, K., & Mohamad, S. (2021). A review on speaker recognition: Technology and challenges. Computers & Electrical Engineering, 90, 107005. https://doi.org/10.1016/j.compeleceng.2021.107005

Hemmat, B. (2024, May 6). How can I perform speaker verification for X-Vectors based on the ivectorSystem documentation? MATLAB Answers. https://www.mathworks.com/matlabcentral/answers/2115001-how-can-i-perform-speaker-verification-for-x-vectors-based-on-the-ivectorsystem-documentation#answer_1453071

Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. Speech Communication, 52(1), 12-40. http://dx.doi.org/10.1016/j.specom.2009.08.009

Microsoft Cognitive Services. (n.d.). Speaker Recognition API - Verification Profile - Create Enrollment. Retrieved June 4, 2024, from https://westus.dev.cognitive.microsoft.com/docs/services/563309b6778daf02acc0a508/operations/56406930e597ed20c8d8549c

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, pp. 5206-5210. https://doi.org/10.1109/ICASSP.2015.7178964

Rabiner, L., & Schafer, R. W. (2007). Chapter 9: Automatic Speech Recognition (ASR). In Introduction to Digital Speech Processing (pp. 176–180).

Reynolds, D.A.; Rose, R.C. (January 1995). "Robust text-independent speaker identification using Gaussian mixture speaker models". IEEE Transactions on Speech and Audio Processing. 3 (1): 72–83. doi:10.1109/89.365379. S2CID 7319345.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2018.8461375

The MathWorks Inc. (2024). Speaker Recognition Using x-vectors. The MathWorks Inc., Natick, Massachusetts. https://www.mathworks.com/help/audio/ug/speaker-recognition-using-x-vectors.html

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(3), 328-339. https://doi.org/10.1109/29.21701