Jurnal Kejuruteraan 36(5) 2024: 1921–1933
https://doi.org/10.17576/jkukm-2024-36(5)-12

Anti-Stammering Algorithm with Adapted Multi-Layer Perceptron

Ali Bashar Hussein[a*], Raid Rafi Omar Al-Nima[a] & Tingting Han[b]

[a]*Department of Medical Instrumentation Technology Engineering, Technical Engineering College of Mosul, Northern Technical University, Ninevah, Iraq*

[b]*School of Computing and Mathematical Sciences, Birkbeck, University of London, London, United Kingdom*

**Corresponding author: e.alibashar@gmail.com*

Received 20 January 2024, Received in revised form 18 May 2024
Accepted 11 July 2024, Available online 30 September 2024

ABSTRACT

Stuttering (or stammering) is a common speech disorder that may continue until adulthood, if not treated in its early stages. In this study, we suggested an efficient algorithm to perform stammering corrections (anti-stammering). This algorithm includes an effective feature extraction approach and an adapted classifier. We introduced Enhanced 1D Local Binary Patterns (EOLBP) for the extraction of features and adapted a classifier of Multi-Layer Perceptron (MLP) neural network for regression. This paper uses a database that involves speech signals with stammering, it can be called the Fluency Bank (FB). The result reveals that the proposed anti-stammering algorithm obtains promising achievement, where a high accuracy of 97.22% is attained.

Keywords: Feature Extraction; Machine Learning; Regression Neural Network; Stammering; Stuttering

INTRODUCTION

As a verbal communication tool, people use speech to express ideas, thoughts (Manjula, Shivakumar, and Geetha 2019). Around the world, different types of speech disorders can be found, such as stammering/stuttering, dysarthria, lisping, cluttering, spasmodic dysphonia, aphasia, mutism and apraxia of speech. Here, we will focus on stammering/stuttering (Prabhu et al. 2020).

When stammering happens, speeches continuity is disrupted by dysfluencies; examples include pauses, prolongations and repetitions (Manjula, Shivakumar, and Geetha 2019). Repetition is when a word is said two or more times (Prabhu et al. 2020). Prolongation is stretching the duration of some strings during the speech (Prabhu et al. 2020). Pause is a significant factor that is considered in stammering if it crosses a specific amount of time (Prabhu et al. 2020). Around 1% of people have a noticeable stammering issue and it has been discovered that the affected female to male ratio is 1:3 or 1:4 times (Manjula, Shivakumar, and Geetha 2019). Examples of a stammered speech signal and a normal speech signal are presented in Figure 1 and Figure 2. The latter contains repetition, prolongation and pause.

The main aim of this study is to adapt the Multi-Layer Perceptron (MLP) algorithm to perform stammering corrections for stammered speeches. This can help people who stutter so that they can communicate and share their ideas in a better way. We summarize the main contributions of this paper below:

1. Discovering the database for stammering/stuttering speeches -- the Fluency Bank (FB).
2. Pre-processing the database.

3. Proposing an efficient anti-stammering algorithm.
a. Introducing a feature extraction method called the Enhanced 1D Local Binary Patterns (EOLBP).
b. Adapting the MLP for regression to address stammering speech issues.

In this paper, the remaining sections are as follows, Section 2 states the materials and methods, Section 3 shows and discusses the results, and Section 4 concludes the paper.

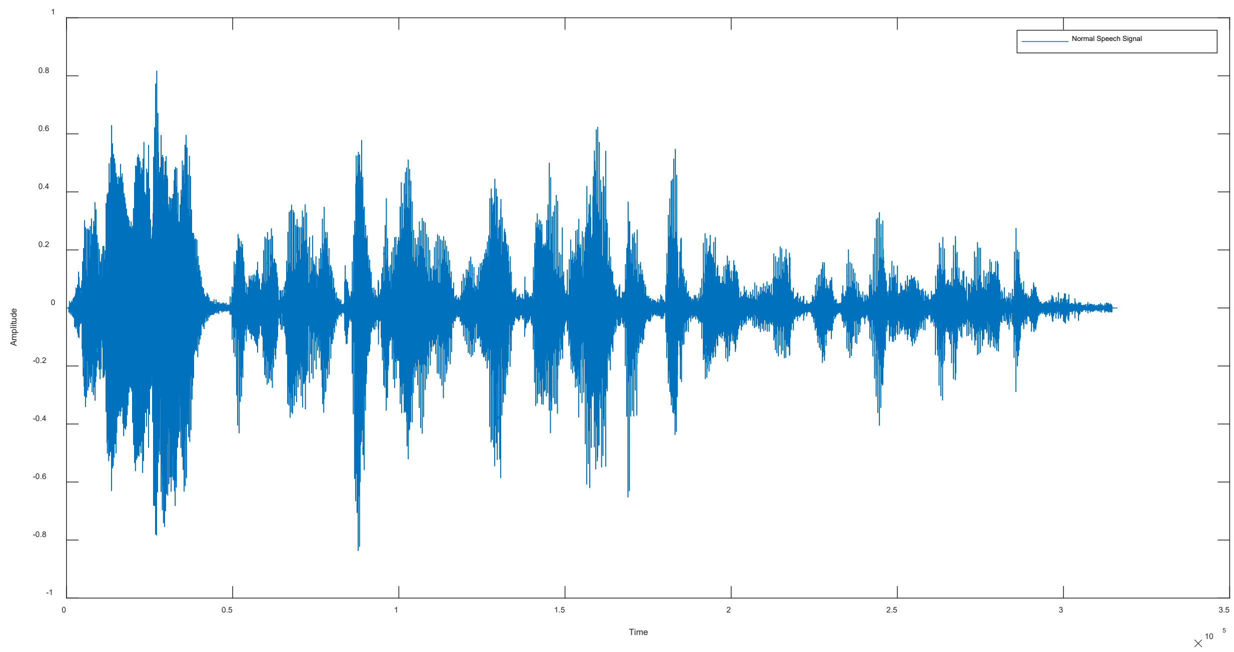


FIGURE 1. Normal speech signal example

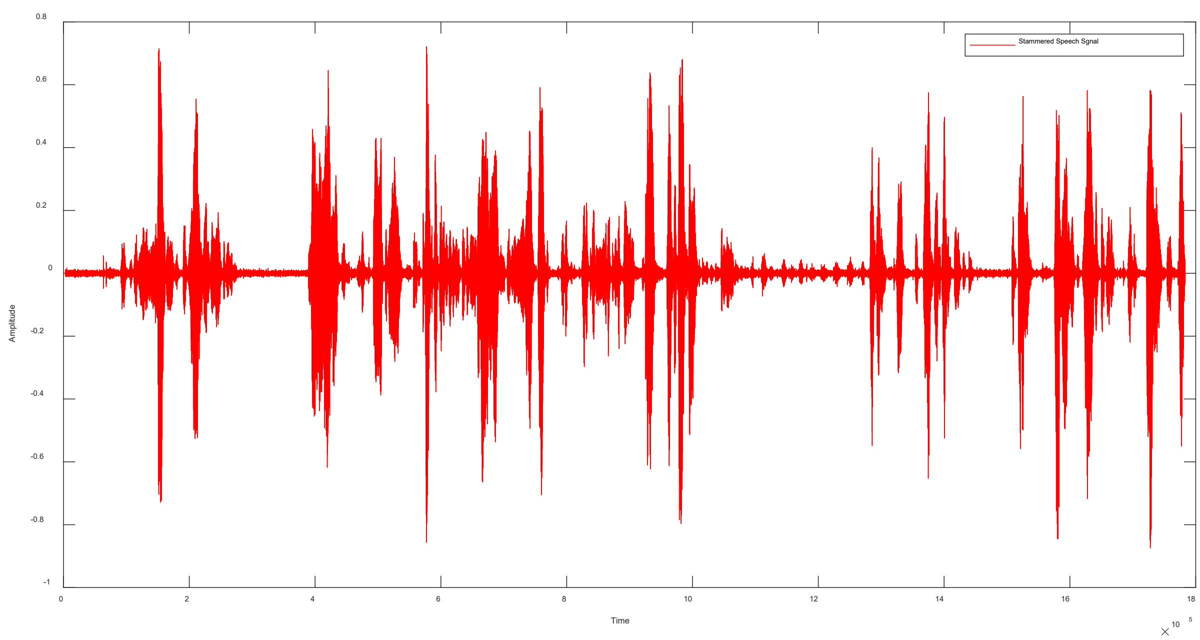


FIGURE 2. Stammered speech signal example

A chronological overview of interesting studies that have been completed in the past few years:

In 2012, Bahadorinejad and Almasganj suggested the Delayed Auditory Feedback (DAF) technique to overcome stammering speech problems. By using the DAF device, a patient hears his/her voice after a short delay. DAF trains patients to talk fluently by playing back the person's voice to them and they hear the voice after a small delay, often approximately 0.1 seconds late. The purpose of this work was to build a DAF to treat stuttering and relieve stress from patients which cause stuttering (Bahadorinejad & Almasganj 2012).

In 2013, Zhang et al. introduced an automatic detection of repeated stuttering with computer assistance to discover Chinese stuttering speech. In order to identify stuttered repetitions with multi-syllables in Chinese speech, the first step was to build a multi-span looping forced alignment decoding network. As a second step, to reduce the mistakes from decoding networks, the authors added a branch penalty factor to the networks to adjust the direction of decoding using recursive search. Finally, to increase the accuracy of detecting result, detected stutters were then re-evaluated using confidence calculations. Experimental results demonstrated that the suggested algorithm could further enhance system performance with a comparatively low average detection error rate of 18% (Zhang et al. 2013).

In 2015, Ramteke et al. worked on finding repetitions in stutterer's speech. Based on energy, stuttering speech signal was split into separated units. Features of formants, shimmer and the Mel Frequency Cepstral Coefficients (MFCCs) were utilized for detecting repetitions. These features were taken from each separated unit. Then, they were compared with the following units within speech period of one second by using the Dynamic Time Warping (DTW). A threshold was established based on analyzing DTW scores and units were considered repeated events if their scores fell below the threshold. In this study, 50 repetition events made up 27 seconds of speech data were used. The outcome demonstrated that repetitions in stuttering speech could be identified using combinations between the MFCCs, formants and shimmer, where 47 of 50 repeats are successfully detected leading to 19 false positive detections with the accuracy of 94% (Ramteke et al. 2016).

In 2016, Surya and Varghese presented a system for automated speech recognition of stuttered people. This work suggested three methods to recognize stuttering in speech: utilizing a trained model, eliminating repetitions and prolongations, and converting to texts. The first method was a supervised model for stuttered speech recognition, where the MFCC feature extraction and Support Vector Machine (SVM) classifier were used. The second method was recognition by stuttering pruning. The third method was automated speech-to-text by using the artificial neural network. Acquired accuracies were 76%, 62% and 80% for the first, second and third methods, respectively (Surya and Varghese B A-B 2016).

In 2018, Dash et al. developed an algorithm for recognizing and correcting stuttered speech to enhance its recognition by using two methods. The first one was for removing prolongation(s) from samples through amplitude thresholding by using a back-propagation neural network. The second one was for removing repetitions using a Text-to-Speech (TTS) system. There were 110 speech samples in total, of which 60 speech samples was utilized for training and 50 speech samples was used for testing which resulted in the accuracy of 86% for the system (Dash et al. 2018).

In 2019, Manjula et al. performed stuttered speech classification using an Adaptive Optimization-Based Artificial Neural Network (AOANN). The aim of this study was to develop an automatic recognition system to evaluate or determine the total counts of repetitions, prolongations and blocks which were stuttering disfluencies. The four stages of the proposed system were: extracting and pre-processing speech signals, applying syllable segmentation by using an autoregressive approach, extracting features by utilizing the MFCC, and dysfluency classifying by employing the AOANN, which is Artificial Fish Swarm Optimization (AFSO) algorithm. As a result, the suggested AOANN approach predicted a count that was similar to the real count for blocks, prolongation and repetition in stuttering (Manjula et al. 2019).

In 2020, Prabhu et al. built an anti-stuttering algorithm which was speech-based by using the Matlab software for removing word repetitions. The implementation of this method consisted of five stages: filtering of magnitude for prolongations removal, ejecting of silence, converting voice-to-text, repeating removal and converting text-to-speech. The number of samples that was used in this work reached to 30, among which 26 samples were successfully corrected. The proposed algorithm's results reported an accuracy of 86% (Prabhu et al. 2020).

In 2020, Arjun et al. suggested in disfluent speech an automatic stuttering adjustment. The authors used MFCC to and LPC (Linear Predictive Coefficients) to extract features. The authors used correlation and short-time energy as the criteria for removing prolongations and repetitions between frames. Long pauses were removed from input voices and transformed to sample rate 22.05 kHz, keeping the natural space between words. The accuracies were 97.5% for long pauses, 94.3% for prolongations and 97.5% for repetitions. By collecting more samples of speech from people who stutter and utilizing the adaptive threshold instead of hard threshold, which removed some non-repeated words, the accuracy and efficacy of this method could be further improved (K N et al. 2020).

In 2021, Sheikh et al. introduced a neural network that incorporates time delays (TDNN) to detect stammering. This stuttering detection method was novel as it was based on deep learning and could recognize and detect different kinds of disfluencies. The TDNN could capture contextual features of disfluent utterances. It was trained using the input features of the MFCC. A database was used, where more than 100 speakers were employed for testing. The results demonstrated that StutterNet outperformed those based on RNNs (residual NN) and Bi-Directional Long Short-Term Memory (Bi-LSTM) in the case of overall average accuracy and Matthews Correlation Coefficient (MCC), which attained 4.69% and 0.03, respectively (Sheikh et al. 2021).

In 2022, Harvill et al. presented stutter detection by using frame level and introduced the objective of frame-level stutter identification to determine the time alignment of stuttering in speech utterance. This approach was evaluated on the task of stuttering correction. This work showed that frame-level stutter detection could be trained by utilizing artificial stutter which was pre-trained for sound and word repetition and also showed that the multiple-instance learning was implemented using max-pooling. From the results, all processed speech were seen as having less stuttering than unprocessed speech. This study suggested that the stutter detection method correctly identified the frames in which stuttering was occurred. The most processed speech showed a slight decreasing in the perceived transcript's match. Both sound and word repetition, but not prolongation, helped in removing more stutters with the pre-trained model (Harvill et al. 2022).

To sum up, there were already quite a lot of research on how to recognize and correct stuttering in the literature. Our work makes a notable contribution for anti-stammering by proposing the EOLBP approach and adapted the MLP for regression. As far as we are aware, no study considered such contributions for addressing stammering speech.

METHODOLOGY

ANTI-STAMMERING ALGORITHM

SUGGESTED ANTI-STAMMERING ALGORITHM

In this study, anti-stammering algorithm have been suggested an to overcome stammering in speeches. This algorithm could also help speech-language pathologists instead of accomplishing boring routine tasks to help people who stutter. The proposed algorithm includes the following stages:

1. Preparing stammering and anti-stammering signals database.
2. Employing segmentation to divide stuttered and anti-stammered signals into sentences.
3. Applying the augmentation process to segmented speech signals for the testing case.
4. Employing the proposed EOLBP approach for extracting features.
5. Using adapted MLP for regression in train and test phases.
6. Keeping the last weights that are obtained from the training phase. After that, applying these weights to the testing phase.
7. Determining the performance of the tested stammering signals according to the provided anti-stammering outputs.

Figure 3 demonstrates the block diagram of the suggested anti-stammering algorithm, it displays its mentioned stages.

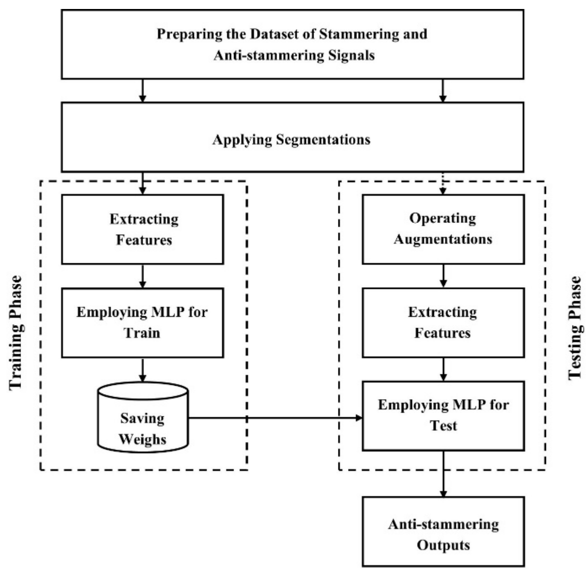


FIGURE 3. Block diagram of the suggested anti-stammering algorithm

PREPARING THE SIGNALS DATABASE OF STAMMERING

This study has explored the FluencyBank (FB) database to be very useful. The FB is an accessible database devoted to improving fluency. Furthermore, it is valuable, reliable and special ("FluencyBank," n.d.). FB is a part of the bigger system of TalkBank (TB) that has recently been funded by the National Institute on Deafness and other Communication

Disorders (NIDCD) and National Science Foundation (NSF) (Bernstein Ratner and MacWhinney 2018).

The preparing includes:

1. Gathering the FB database, which contains videos of 24 stuttered speeches and 1 anti-stuttered speech for reading united lengthy texts by participants of diverse ages.
2. Extracting the stuttered speech signals (voices) for every participant. As a result, 24 stuttered signals are obtained in the format (mp3).
3. Extracting the signal (voice) of anti-stuttered speech from the anti-stuttered video to be employed as targets, which they are also prepared in the format (mp3).

Preparation of segmentation includes:

1. Segmenting every participant's stuttered voice into small sentences, producing 360 segmented voices for all of the participants.
2. Segmenting every participant's anti-stuttered voice into small sentences, producing 15 segmented voices (targets) for every participant.
3. Equalizing the sizes of segmented signals to ensure they all have the same size and they all are ready to be employed in the remaining algorithm parts.

FEATURE EXTRACTION APPROACH

In this study, the conventional LBP method has been enhanced in order to approach the EOLBP feature extraction. It takes into account 8 neighbor values for a portion of the 1D stuttering speech signal that are positioned horizontally around a center value. This configuration can afford acceptable performance as confirmed in (Liu, Tian, and Ma 2013).

The following steps demonstrate how to compute the feature extraction of the EOLBP for a 1D vector (signal):

1. Taking into consideration a window with size 9 values at the beginning of 1D vector. It involves a center value and its neighbor values, where the neighbor values are organized as four values at the right of the center value and four values at the left of the center value.
2. Considering that the threshold is the center value of the window.
3. Comparing every neighbor value with the threshold in the same window. Logic 1 will be used if the threshold is lower than or equal the neighbor value. If not, logic 0 will be used.
4. Transferring the eight logical numbers to their equivalent decimal number.
5. Shifting the window one value to the right along the 1D vector and reiterating steps 2 through 4 until the entire 1D vector length is considered.
6. Applying divisions for the corresponding values of the original signal out of the calculated decimal numbers.

It should be noted that steps 3 and 4 correspond to equations 1 and 2. In addition, step 6 aims to maintain the variations of the original signal. Figure 4 displays an illustration of essential processes in the proposed EOLBP technique.

MLP FOR REGRESSION

PRINCIPLES OF REGRESSION

Regression is a technique that utilizes to provide a relationship between dependent variables and independent variables (Kavitha S, Varuna S, and Ramya R 2016). It has the capability to address complex problems. It expects the outcomes of dependent variables depending on independent variables. Principally, relationship of regression may be non-linear or linear (Seber and Lee 2003; Khuri 2013; Kavitha S, Varuna S, and Ramya R 2016). Moreover, linear regression may be of a multiple or simple type. The simple type refers to a relationship of regression between two single variables dependent and independent. Whereas, the multiple type points to a relationship of regression between variables for multiple independent and a variable for single dependent, which is related to our work in this paper.

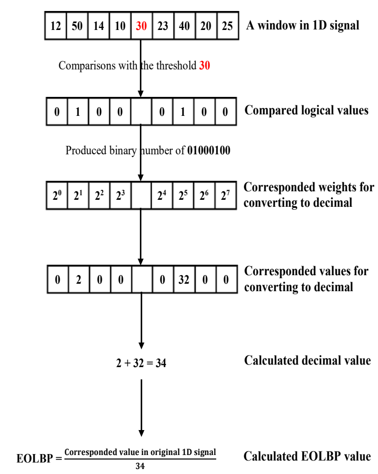


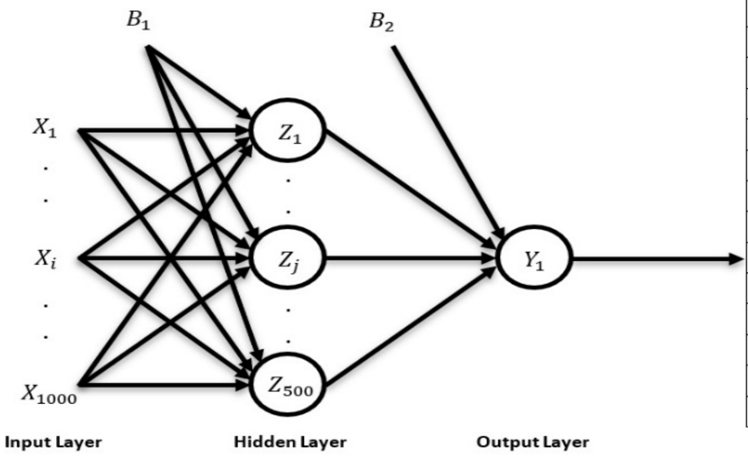
FIGURE 4. An illustration of operations in the proposed EOLBP technique

ADAPTED MLP FOR REGRESSION

The adapted MLP for regression consists of the following characteristics:

1. EOLBP input vectors, where any input vector composes of 1000 values.
2. targets of indices (1, 2, …, 15), each index leads to an assigned anti-stammering sentence.
3. any input vector signifies a stammering sentence feature extraction.
4. single hidden layer of 500 neurons or nodes.
5. single adapted MLP is determined for a stammered user.

Figure 5 shows the architecture of the adapted MLP for regression. It considers two phases: a training phase and a testing phase. During the training phase, the MLP acquires the training EOLBP input vectors of stammering signals. It leans to generate outcome values as the equivalent targets. Target values of indices can lead to their assigned anti-stammering signals. For the testing phase, the augmentation is exploited to provide stammering signals with reasonable changes. Applying augmentation can also enlarge the number of stammering signals. In this study, adding noise stammering voices are used as an augmentation. This is because it is estimated that stammered speech signals are usually combined with noise.



Index	Anti-stammering Signals
1	1st anti-stammering signal
2	2nd anti-stammering signal
3	3rd anti-stammering signal
4	4th anti-stammering signal
5	5th anti-stammering signal
6	6th anti-stammering signal
7	7th anti-stammering signal
8	8th anti-stammering signal
9	9th anti-stammering signal
10	10th anti-stammering signal
11	11th anti-stammering signal
12	12th anti-stammering signal
13	13th anti-stammering signal
14	14th anti-stammering signal
15	15th anti-stammering signal

Lookup Table

FIGURE 5. The architecture of the adapted MLP for Regression

RESULTS AND DISCUSSION

THE FLUENCYBANK DATABASE

Our work is based on the FluencyBank database. The database was volunteered and collected by adult members of National Stuttering Association (NSA) in order to raise a better understanding of the cognitive and emotional characteristics of the people who stutter and the actions to take to live with the stuttering adults ("Teaching with FluencyBank," n.d.).

Since June 2017, the FB database was established with 25 adult contributors who had different degrees of stuttering and all had been diagnosed by University of Maryland's institutional review board (PI: Nan Bernstein Ratner). There were 16 male and 9 female participants who were in the range of 24–62 years old. These participants were put into a clinical group and were asked to was read out a Friuli passage which contains 369 syllables in total from Stuttering Severity Instrument-4 (SSI4). All the readings were recorded in .mp4 format. The sampling rate of 22 participants' speeches was 48000 kHz, whereas it was 44100 kHz for the other 3 participants ("Teaching with FluencyBank," n.d.).

IMPLEMENTATION AND RESULTS

Our algorithm was implemented in MATLAB. The implementation consisted the training phase and testing phase. We are to present the implementation details and the results from each phase below.

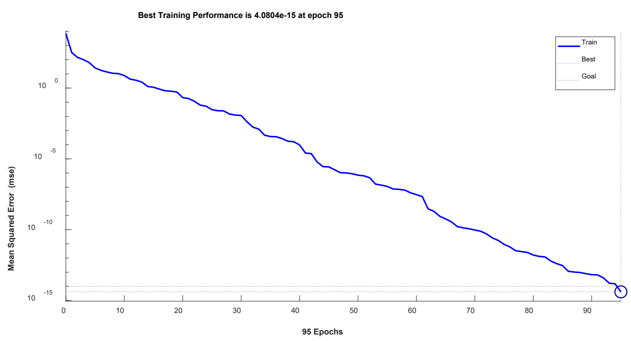
TRAINING IMPLEMENTATION

The input of our algorithm is speeches with stuttering. The first step is to apply the EOLBP feature extraction. We then used the adapted MLP process to perform regression.

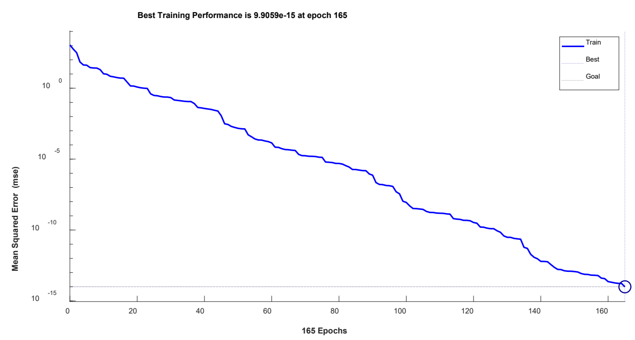
For each sentence with stuttering, we use 1000 nodes (the same number as in the feature extraction) in the MLP input layer. We also set 500 nodes to be in the hidden layer of the MLP and only one node in the output layer, as this is the regression target. After this, we apply the tan sigmoid transfer function to the MLP hidden layer and pure linear function to the output layer. For training the MLP, Scaled Conjugate Gradient (SCG) type is chosen. Furthermore, the maximum epochs number is set to 1000. Meanwhile, minimum Mean Square Error (MSE) of 1×10-14 is used. Finally, we perform the EOLBP feature extraction for the stuttering sentences and we utilize one MLP for each person.

TRAINING RESULTS

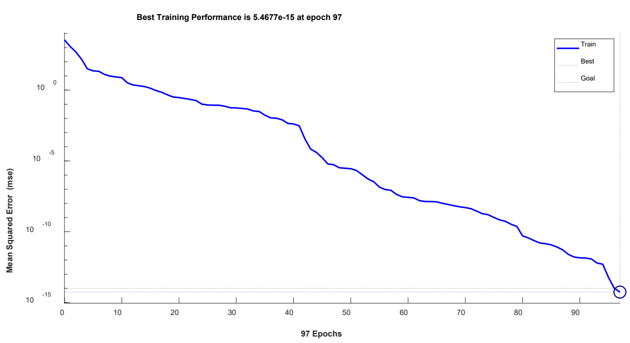
We present some results of training the MLP for stammering speeches in Figure 6. We notice that the mean squared errors were far from satisfactory at the beginning of each process. As the epochs went on, the weights had updated many times and the MSEs reduced steadily, so the curve declined all the way. All these curves indicated that the training processes were successful.



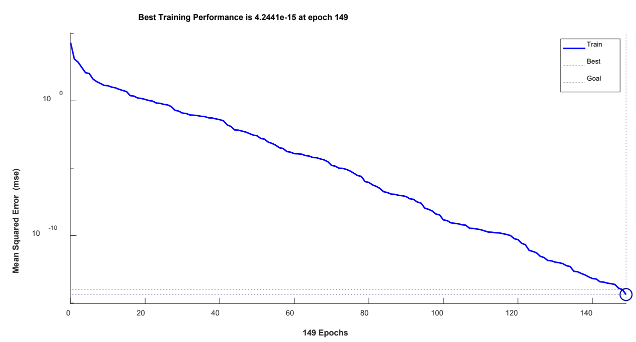
(i)



(ii)



(iii)



(iv)

FIGURE 6. The MSE curves for training the MLP with stuttering speech signals(i) to (iv) are for four different participants

In Figure 7, we plotted the provide relationships between the outputs of MLP and the desired targets after the linear regression to provide anti-stammering speech signals. The linear regression plots again indicate that the training was effective as all the data lie perfectly on the best-fitted lines. This was supported by the R value R=1, which was the best it could achieve. The result was indeed very promising in our adapted MLP training phase. In the rest of this chapter, we will show the results in the testing phase.

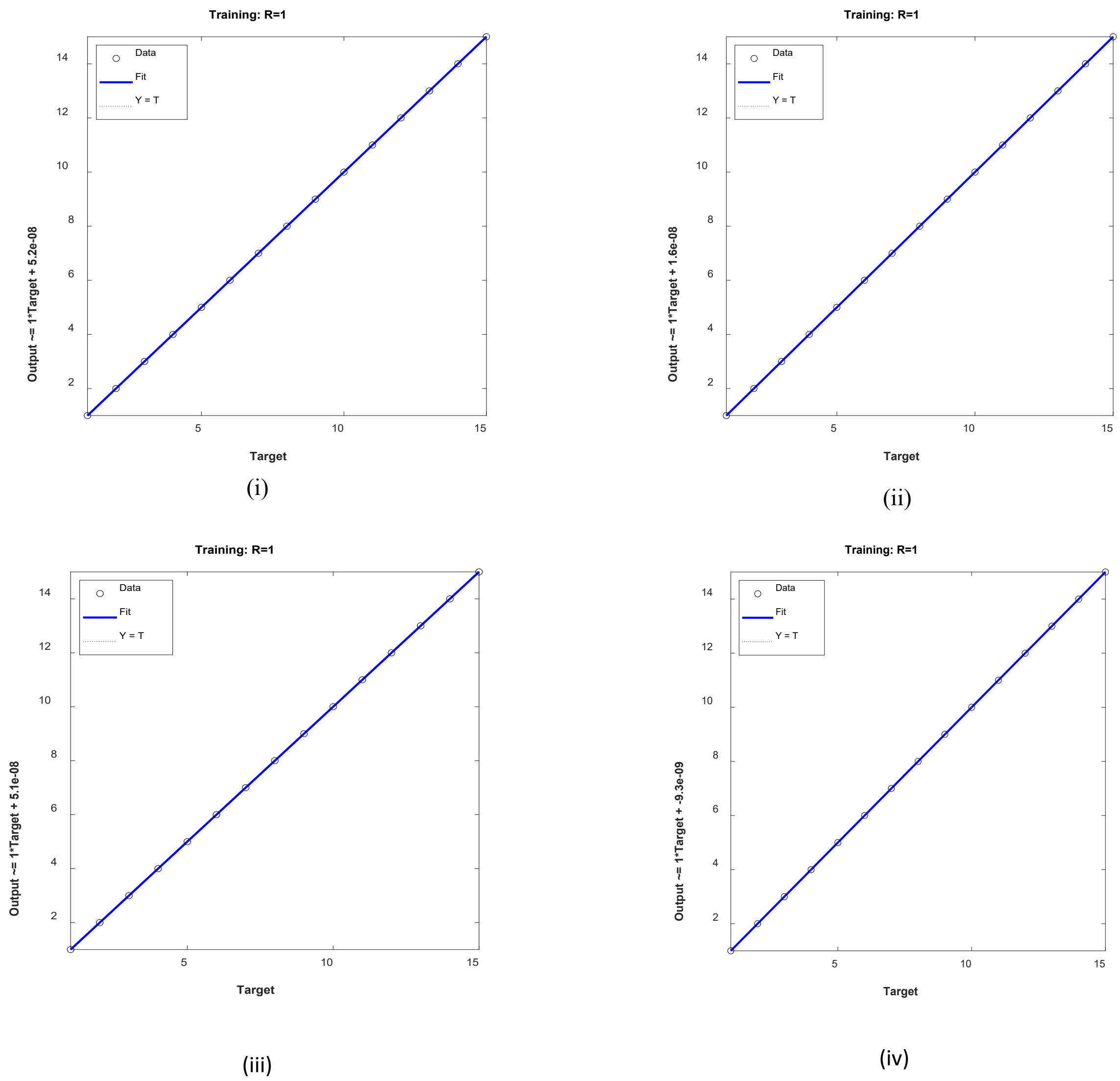
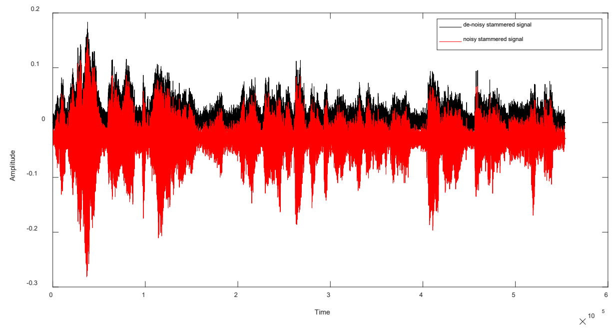


FIGURE 7. Adapted MLPs for regression - linear regression results(i)-(iv) are for four different participants

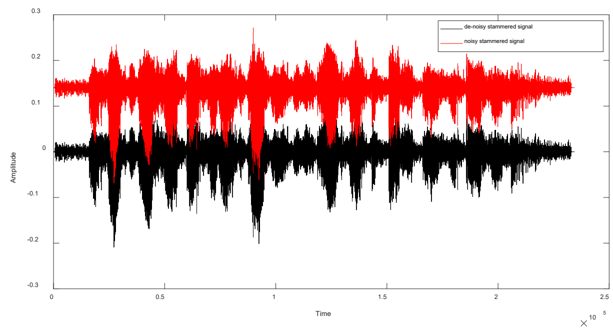
AUGMENTATION IN TESTING

We use augmentation for the stuttered speech signals in the testing phase. More specifically, random noises are added to the speech signals before implementing the EOLBP feature extraction and adapted MLP. There are two reasons to apply augmentation. On one hand, we can increase the number of samples in the database. This kind of increase is particularly valuable when the original database is not on a large scale. On the other hand, it is natural to have noises in the speech signals, no matter with or without stammering. The noises normally come from either the environment or the devices that make the recordings, or both.

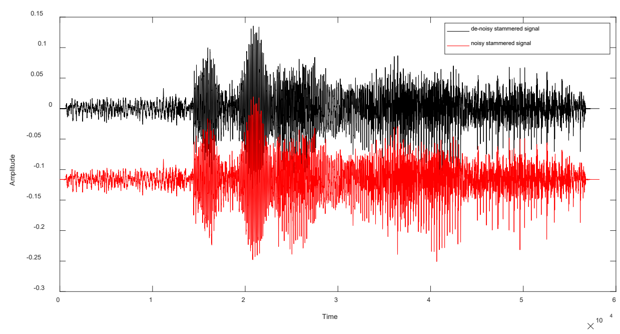
The way to implement augmentation is to produce normally distributed random values alongside the stammering speech signals. We present a comparison between the stuttering speech signals before and after augmentation for four different example signals in Figure 8.



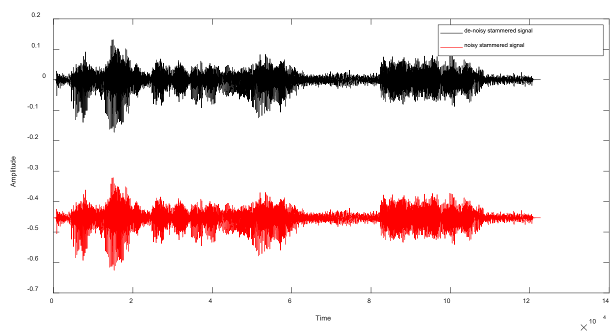
(i)



(ii)



(iii)



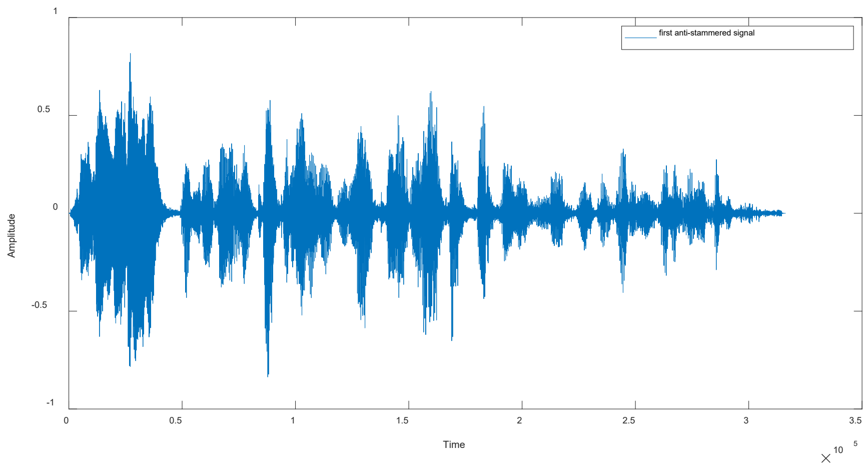
(iv)

FIGURE 8. Comparison of the stuttering speech signals before and after augmentation (i)-(iv) are the first four signals from the first participant

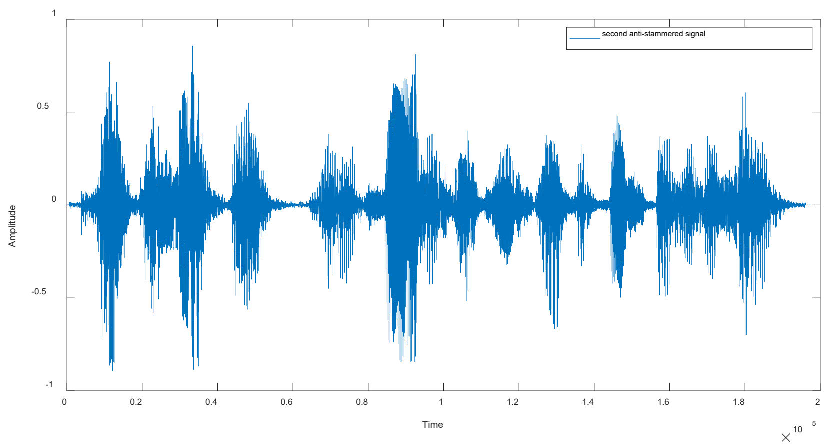
TESTING RESULTS

Each augmented signal is then passed on to the EOLBP feature extraction part before using the adapted MLP to generate a pure anti-stuttering signal. Four example output stammer-free speech signals are presented in Figure 9.

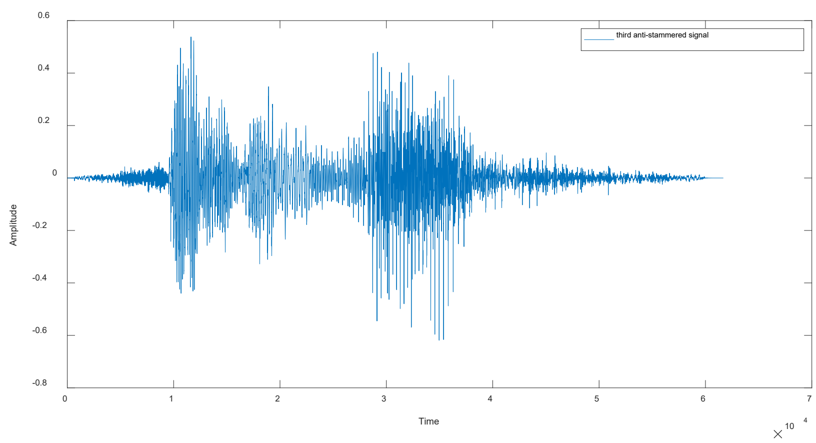
As the testing result, our adapted MLP algorithm has achieved a very high regression accuracy of 97.22% and the error rate is 2.78%.



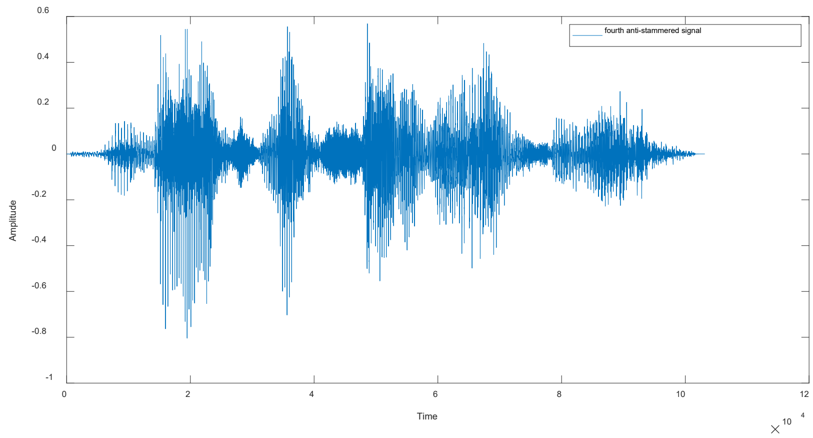
(i)



(ii)



(iii)



(iv)

FIGURE 9. The speech signals with stammering removed(i)-(iv) are the first four signals from the first participant

COMPARISON WITH OTHER NEURAL NETWORKS

In order to compare the performance of our adapted MLP with other neural networks, such as (R Al-nima 2010; Raid Rafi Al-Nima, Al-Ridha, and Abdulraheem 2019; Al-Kaltakchi et al. 2019; Raid R Al-Nima 2012), we will use the same anti-stammering procedure, and only change the adapted MLP for other neural networks. This way can the comparison be fair. We present in Table 1 all the accuracy and error rate results of our adapted MLP, generalized Regression Neural Network (GRNN), Exact Radial Basis Neural Network (ERBNN), Radial Basis Neural Network (RBNN) and Cascade-Forward Neural Network (CFNN).

TABLE 1. Performance comparison: same anti-stuttering procedure, different neural networks

Method	Accuracy	Error
The adapted MLP	97.22%	2.78%
CFNN	92.77%	7.23%
ERBNN	91.94%	8.06%
RBNN	68.88%	31.12%

Table 1 shows that GRNN is the least effective as its error rate is very high 93.34%. The RBNN performs much better than GRNN accuracy 68.88%, but still not satisfactory. The ERBNN and CFNN perform well and achieve a similar accuracy of 91.94% and 92.77%, respectively. Our adapted MLP reported the best accuracy of 97.22% and is the most suitable neural network model for the anti-stuttering algorithm.

COMPARISON WITH OTHER APPROACHES

Another comparison is made between our adapted MLP for regression and other approaches in (Dash et al. 2018), (Prabhu et al. 2020) and (K N et al. 2020).

TABLE 2. Comparison between our adapted MLP for regression and other approaches

Reference	Method	Stuttering Dysfluencies	Accuracy
(Dash et al. 2018)	Backpropagation algorithm	Prolongation Repetition Interjections	86%
(Prabhu et al. 2020)	Implemented anti-stuttering algorithm	Repetitions Prolongations Silence Ejection	86%
(K N et al. 2020)	Signal processing method	Repetition	88.35%
		Prolongation	94.3%
		Long pauses	97.5%
This paper	Adapted MLP for regression	Any stammering type(s)	97.22%

Table 2 shows the methods, the stuttering dysfluencies to be tackled and the accuracy obtained. It can be seen that (Dash et al. 2018) has obtained the accuracy of 86% by using backpropagation algorithm. In (Prabhu et al. 2020), the same accuracy has been achieved. In (K N et al. 2020), better accuracies of 88.35%, 94.3% and 97.5% have been attained for repetition, prolongation and long pauses, respectively. Finally, the adapted MLP has benchmarked a high accuracy of 97.22% for any stammering type(s). Our method has attained the performance of 97.22% for the whole stuttering dysfluencies, whereas the result of 97.5% is achieved in (K N et al. 2020) for just the long pauses. The accuracies are lower than ours for other stammering types of repetition and prolongation in (K N et al. 2020)

The purpose of using the applied methods such as the EOLBP features extraction and the adapted MLP for regression is that they achieved our target of extracting the features precisely and provide the anti-stammering signals perfectly. It appears that the proposed anti-stammering approach proves its strengths from the final results which approved our point of view about the reasons behind employing these applied methods.

RESULTS DISCUSSIONS

The training phase results show that the trainings are successful since their performance curves are descending toward a low error value. Also, the linear regression curves provide another indicator of the trainings' success since all of the related data are appropriately situated on the best-fit lines. In the testing phase, a very high accuracy and a very low error are benchmarked as 97.22% and 2.78%, respectively. Such results refer to the acceptability of suggested anti-stammering algorithm. The comparisons also showed how our work can surpass and overcome with other networks and approaches.

Multiple advantages are highlighted for our proposed anti-stammering algorithm as follows:

1. stammering curing can be provided.
2. any stammering type(s) can be considered.
3. pure anti-stammering signals can be reached.
4. high performances are attained.

CONCLUSION

We have worked on stammering corrections, which may help people who stutter, as they can communicate and share their ideas freely without feeling stress or shy towards others. We proposed an efficient anti-stammering algorithm. More specifically, we used EOLBP to extract features from one-dimensional stutter signals and adapted the MLP models for regression to provide anti-stammering signals. We explored the FB database. Among the 720 stuttered sentence signals, 360 signals were used in the training and the remaining 360 augmented signals were for the testing. In addition, we also used 15 anti-stuttering sentence signals from the FB database as targets. The best overall accuracy of our proposed anti-stuttering algorithm is 97.22% and the lowest error is 2.78%.

The obtained results are highly acceptable. They are being in that manner because of two reasons. Firstly, employing the suggested EOLBP features extraction which has the ability to extract effective features from stammered speech signals. Secondly, because of the adapted MLP for regression. This method has the privilege of providing very precise results in the testing phase, where the targets are indexed values, and each refers to a certain anti-stammering sentence. So that whenever the training is successful in the training phase, the output can be precise in the testing phase.

ACKNOWLEDGEMENT

The authors would like to thank University of Technology and Applied Sciences, Oman for supporting the research.

DECLARATION OF COMPETING INTEREST

None.

REFERENCES

Al-Kaltakchi, Musab T S, Raid Rafi Omar Al-Nima, Mohammed A M Abdullah, & Hikmat N Abdullah. 2019. Thorough evaluation of TIMIT database speaker identification performance under noise with and without the G. 712 type handset. *International Journal of Speech Technology* 22: 851–863.

Al-Nima, Raid R. 2012. Human authentication with earprint for secure telephone system. *Iraqi Journal of Computers, Communications, Control and Systems Engineering IJCCCE* 12 (2): 47–56.

Al-Nima, Raid Rafi, Moatasem Yaseen Al-Ridha, & Farqad Hamid Abdulraheem. 2019. Regenerating face images from multi-spectral palm images using multiple fusion methods. *Telecommunication Computing Electronics and Control (Telkomnika)* 17 (6): 3110–3119.

Bahadorinejad, Arghavan, & Farshad Almasganj. 2012. Delayed Auditory Feedback for Speech Disorders. *2012 International Conference on Biomedical Engineering (ICoBE)*, 585–588.

Bernstein Ratner, Nan, & Brian MacWhinney. 2018. Fluency Bank: A new resource for fluency research and practice. *Journal of Fluency Disorders* 56 (June): 69–80.

Dash, Ankit, Nikhil Subramani, Tejas Manjunath, Vishruti Yaragarala, & Shikha Tripathi. 2018. Speech Recognition and Correction of a Stuttered Speech. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1757–1760.

FluencyBank. n.d. Accessed June 1, 2023. https://fluency.talkbank.org/.

Harvill, John, Mark Hasegawa-Johnson, & Chang D. Yoo. 2022. Frame-Level Stutter Detection. *International Speech Communication Association* (*Interspeech) 2022*, 2843–2847.

K N, Arjun, Karthik S, Kamalnath D, Pranavi Chanda, & Shikha Tripathi. 2020. Automatic correction of stutter in disfluent speech. *Procedia Computer Science* 171: 1363–1370.

Kavitha S, Varuna S, & Ramya R. 2016. A comparative analysis on linear regression and support vector regression. *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, 1–5.

Khuri, André I. 2013. *Introduction to Linear Regression Analysis*, Fifth Edition by Douglas C. Montgomery, Elizabeth A. Peck, & G. Geoffrey Vining. International Statistical Review.

Liu, Ming, Yongmei Tian, & Yinghui Ma. 2013. Inner-Knuckle-Print Recognition Based on Improved LBP. *Proceedings of the 2012 International Conference on Information Technology and Software Engineering: Software Engineering & Digital Media Technology*, 623–630.

Manjula, G., M. Shivakumar, & Y. V. Geetha. 2019. Adaptive Optimization Based Neural Network for Classification of Stuttered Speech. *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 93–98.

Prabhu, R., P. Praveen Kumar, J. Kavin kumar, & Dhivyapriya E. L. 2020. Speech based anti stuttering algorithm using Matlab. *International Journal of Engineering and Advanced Technology,* 9 (3): 2152–2155.

R Al-nima, Raid. 2010. Design a biometric identification system based on the fusion of hand geometry and backhand patterns. *IRAQI Journal of Statistical Sciences,* 10 (1): 169–180.

Ramteke, Pravin B., Shashidhar G. Koolagudi, & Fathima Afroz. 2016. Repetition Detection in Stuttered Speech. *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, Smart Innovation, Systems and Technologies*, 611–617.

Seber, George A. F., & Alan J. Lee. 2003. *Linear Regression Analysis*. John Wiley & Sons, Inc.

Sheikh, Shakeel A., Md Sahidullah, Fabrice Hirsch, & Slim Ouni. 2021. StutterNet: Stuttering Detection Using Time Delay Neural Network. *2021 29th European Signal Processing Conference (EUSIPCO)*, 426–430.

Surya, Arya A, & Surekha Mariam Varghese B A-B. 2016. Automatic speech recognition system for stuttering disabled persons. *International Journal of Control Theory and Applications*. 9 (43): 269–278.

Teaching with FluencyBank. n.d. Accessed May 25, 2023. https://fluency.talkbank.org/teaching/AWS.html.

Zhang, Junbo, Bin Dong, & Yonghong Yan. 2013. A Computer-Assist Algorithm to Detect Repetitive Stuttering Automatically. *2013 International Conference on Asian Language Processing*, 249–252.