StutterNet: Stuttering Detection Using Time Delay Neural Network

Shakeel A. Sheikh[1], Md Sahidullah[1], Fabrice Hirsch[2], Slim Ouni[1]
[1]*Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France*
[2]*Université Paul-Valéry Montpellier*, CNRS, Praxiling, Montpellier, France
[1]{shakeel-ahmad.sheikh, md.sahidullah, slim.ouni}@loria.fr, [2]fabrice.hirsch@univ-montp3.fr

***Abstract*—This paper introduces *StutterNet*, a novel deep learning based stuttering detection capable of detecting and identifying various types of disfluencies. Most of the existing work in this domain uses automatic speech recognition (ASR) combined with language models for stuttering detection. Compared to the existing work, which depends on the ASR module, our method relies solely on the acoustic signal. We use a time-delay neural network (TDNN) suitable for capturing contextual aspects of the disfluent utterances. We evaluate our system on the *UCLASS* stuttering dataset consisting of more than 100 speakers. Our method achieves promising results and outperforms the state-of-the-art residual neural network based method. The number of trainable parameters of the proposed method is also substantially less due to the parameter sharing scheme of TDNN.**

***Index Terms*—stuttering, speech disfluency, speech disorder, time delay neural network.**

I. Introduction

The *speech disorders* problem refers to the difficulties in the production of speech sounds. The various speech disorders include *cluttering*, *lisping*, *dysarthria*, *stuttering*, etc. Of these speech disorders, stuttering – also known as stammering – is the most predominant one [1]. About 70 million people that comprise 1% of the world population suffer from stuttering [2]. People with the stuttering problem face several difficulties in social and professional interactions. This work is about the automatic detection of stuttering with several important applications. For example, it could facilitate the speech therapist's work, since they have to carry out a manual calculation to evaluate the severity of stuttering; to give a feedback to persons who stutter (PWS) about their fluency. Nevertheless, fluent voice is an important requirement for several professions such as news anchoring, emergency announcement, etc. Furthermore, the automatic speech recognition (ASR) system used in voice assistants can be adapted efficiently for PWS.

Even though there are plenty of potential applications, stuttering detection has received less attention, especially from a signal processing and machine learning perspective. Stuttering is a neuro-developmental speech disorder, defined by an abnormally persistent and duration of stoppages in the normal forward flow of speech, which usually takes the form of *core behaviors*: prolongations, blocks, and syllables, words or phrase repetitions [1]. These impact the acoustic properties of speech which can help to discriminate from fluent voice. Studies show that different formant characteristics such as *formant transitions*, *formant fluctuations* are affected by stuttering [1]. The existing methods for stuttering detection employ spectral features such as *mel-frequency cepstral coefficients* (MFCCs) and *linear prediction cepstral coefficients* (LPCCs) or their variants that capture that formant-related information. Other spectral features such as pitch, zero-crossing rate, shimmer, and spectral spread are also used. Finally, those features are modeled with statistical modeling methods such as *hidden Markov model* (HMM), *support vector machine* (SVM), *Gaussian mixture model* (GMM), etc [3].

An alternative strategy of stuttering detection is to apply ASR on the audio speech signal to get the spoken texts and then to use language models [4]–[6]. Even though this method of detecting stuttering has achieved encouraging results and has been proven effective, the reliance on ASR makes it computationally expensive and prone to error.

In this work, we use a deep neural network (DNN) for stuttering detection directly from the speech. In recent decades, the DNNs are widely used in different speech tasks such as speech recognition [7], speaker recognition [8], emotion detection [9], voice disorder detection [10]. However, a little attention has been devoted to the field of stuttering detection.

We propose a *time-delay neural network* (TDNN) architecture for stuttering detection. TDNN has been widely used for different speech classification problems such as speech and speaker recognition [11], [12]. We introduce this for stuttering detection task. The proposed method, referred to as *StutterNet* is a multi-class classifier with output as stuttering types and fluent. Our experiments with the UCLASS dataset show promising recognition performance. We further optimize the *StutterNet* architecture, and we achieved substantial improvement over the competitive DNN-based method.

II. Related Work

The earlier studies in neural network based stuttering detection explored shallow architecture. Howell *et al.* [13], [14] employed two separate *artificial neural networks* (ANNs) for the identification of repetition and prolongation disfluencies. This work used autocorrelation features, envelope parameters, and spectral information input to the neural network. The experiments were conducted with a dataset of 12 speakers. Ravikumar *et al.* [15] attempted *multilayer perceptron* (MLP) for the detection of repetition disfluencies. They used MFCC as input features from 12 disfluent speakers. I. Szczurowska *et al.* employed Kohonen network and MLP for discriminating fluent and disfluent speech [16]. The Kohonen network reduced the dimensionality of the Octave filter-based input

feature. The features were used as an input to the MLP classifier. The experiments were conducted with eight speakers. B. Villegas *et al.* [17] proposed a respiratory-based stuttering classifier. They trained MLP on the respiratory air volume and pulse rate features for the detection of block stuttering. The network is trained on 68 Latin American Spanish speakers. The work in [18] used adaptive optimization based neural network for three class stuttering classification.

Due to recent advancements in deep learning, the improvement in speech technology surpasses the shallow neural network based approaches, and thus, resulted in a shift towards deep learning based framework and, disfluency identification is no exception. The work in [19] used *deep belief networks* with cepstral features for the detection of repetitions and stop gaps on TORGO dataset. T. Kourkounakis *et al*. [20] introduced a deep residual neural network and bi-directional long term short memory (ResNet+BiLSTM) based method to learn stutter-specific features from the audio. They addressed the stuttering detection problem as a multiple binary classification problem. They trained the same proposed architecture for each class of stuttering separately. The method was trained on 24 speakers from the UCLASS [21] stuttering datatset, and considered spectrograms as input feature. The learned features from residual blocks were fed to two bi-directional recurrent layers to capture the temporal context of the disfluent speech.

Although this method has shown promising results in stuttering detection, it has several limitations. First, this method did not consider fluent speech, and the experiments are performed within stuttering classes. Second, the technique requires training of multiple models for each type of disfluency. Third, the model has a huge number of parameters ($\approx$24 million), thus makes it computationally expensive to train. Furthermore, the experiments are conducted with only a small subset of speakers.

In this paper, we address the above-mentioned problems with *StutterNet* based on TDNN. This type of architecture is suitable for speech data as it captures temporal convolution as well as captures contextual information for a given context [11]. We address stuttering type detection as a multi-class classification problem by training a single *StutterNet* including data from all types of stuttering. Due to the parameter sharing in TDNN, we significantly reduce the number of parameters. In the next section, we provide the details of our proposed architecture.

III. Proposed Architecture

As discussed in Section I, due to the very limited research in the field of stuttering detection, the idea is to design a single network that can be used to detect and identify various types of stuttering disfluencies. The proposed *StutterNet* first computes the MFCC features from audio samples, which are then passed to the TDNN [12] to learn and capture the temporal context of various types of disfluencies.

A. Acoustic features

In developing any speech domain application, the representative feature extraction is the most important that affects the model performance [20], [22], [23]. With the aid of signal processing techniques, several features of the stuttered speech signal can be extracted like raw waveform, spectrograms, mel-spectrograms, and or MFCCs. However, our aim is to compute and extract the features that compactly characterize the stuttering embedded in a speech segment and also which approximates the human auditory system's response. For stuttering domain, MFCCs are the best suitable and are the most commonly used features in stuttered speech domain [24], thus, we use MFCCs as the sole features to our *StutterNet* network. These features are generated after every 10 ms on a 20 ms window for each 4 sec audio sample. This four-second window is used as stuttering lasts on average four seconds [1].

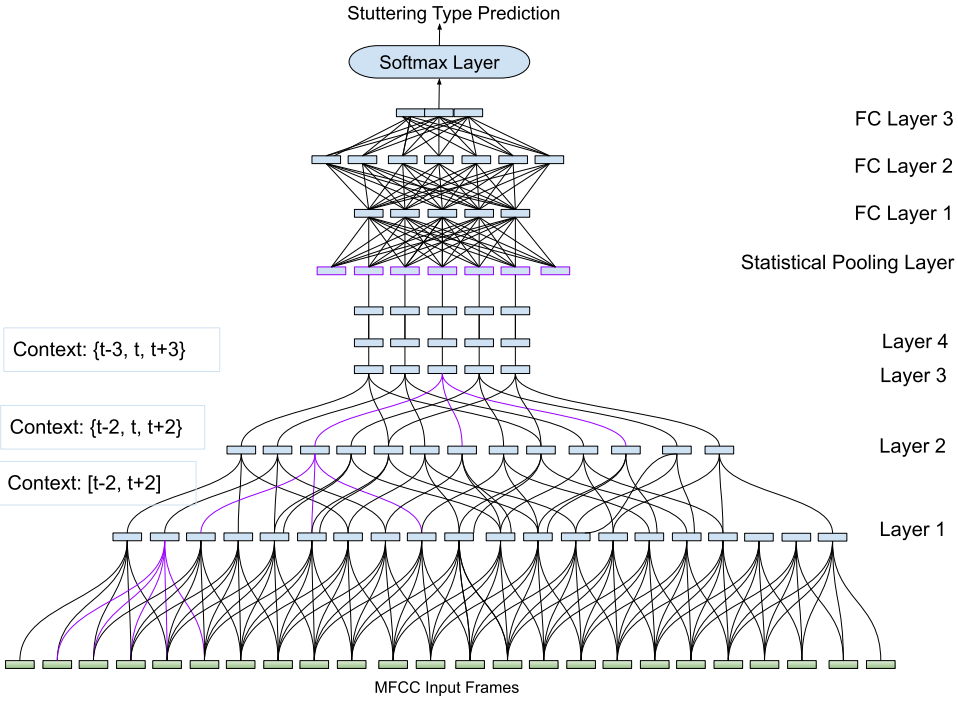
B. StutterNet architecture

Most of the existing work in literature has studied the stuttering detection as a binary classification problem: stuttering versus fluent detection [3] or one type versus other disfluency types [20]. We tackle the stuttering detection as a multi-class problem of detecting and identifying the core behaviors, as opposed to the work done in [20], who addressed this problem as multiple binary classification, with the same network used for every disfluency. For this multi-class detection, we propose a TDNN [11] based *StutterNet* which effectively learns stutter-specific features. The TDNN method is well suited in capturing the temporal [11], [12] and contextual aspects of various types of disfluencies. The neural network takes 20 MFCCs as an input features to learn and capture the temporal context of stuttering. The *StutterNet* contains five time delay layers with the first three focusing on the contextual frames of $[t-2, t+2], \{t-2, t, t+2\}, \{t-3, t, t+3\}$ with dilation of 1, 2 and 3 respectively. This is followed by statistical pooling, three fully connected (FC) layers and a softmax layer that reveals the prediction of multiclass stuttering disfluencies. Each layer is followed by a ReLU activation function and 1D batch normalization except statistical pooling layer. A dropout of 0.2 is applied to first two fully connected layers. The model architecture is shown in Fig.1.

IV. Experimental Evaluation

A. Dataset

We have used the UCLASS release 1 stuttering dataset that has been created by the Department of Psychology and Language Sciences, University College London [21]. This dataset consists of monologue samples from 139 participants aged between 5$-$45 years. Of these, 128 have been chosen in this case study with females 18 and males 110. Of these, 104 speakers were used for training, 12 for validation and 12 for testing. The audio samples were annotated manually by listening to the recordings of speech segments. The annotations have been carried out into different categories of stuttering including: core behaviors, fluent, repetition$-$prolongations, blocks$-$prolongations, repetition$-$blocks, and blocks$-$repetition$-$prolongations. However, in this paper, we are focusing only on the core behaviors as the dataset contains only few of the remaining



StutterNet with input: 20 (MFCC dim) X *T* (Chunk size)

Layer	Input X Output	Context
TDNN1	20 X 512	[t-2, t+2]
TDNN2	1536 X 512	{t-2, t, t+2}
TDNN3	1536 X 512	{t-3, t, t+3}
TDNN4	512 X 512	{t}
TDNN5	512 X 1500	{t}
Statistical Pooling	1500*T* X 3000	[0, *T*)
FC1	3000 X 512	-
FC2	512 X 512	-
FC3	512 X *N*	-

Fig. 1: *(a)*: StutterNet layers and context-wise computation, *(b)*: *StutterNet* Architecture (Baseline) (Except statistical pooling layer, each layer is followed by a ReLU activation function and batch normalization)

ones. Each monologue audio clip was sliced into 4-second and down sampled to 16 *kHz* segments, resulting in a total of 4674 speech segment samples. Due to the lack of standard disfluent speech data, we have used only UCLASS dataset for our experimental studies.

In order to evaluate the *StutterNet* method on the UCLASS dataset, we adopted *K*-fold cross validation technique , where *K=10*. We conducted 10 experiments, each consisting of random sampling of 80% for training, 10% for validation and last 10% for testing. The reported results are the average between 10 experiments. All experiments were trained with an early stopping criteria of patience 7 on validation loss.

B. Evaluation metrics

In order to evaluate the model performance on this UCLASS dataset, we have used the metrics including: precision, recall, F1-score and accuracy which are the standard and are widely used in the disfluent speech domain [20]. In addition to these, we choose Matthew's correlation coefficient (MCC), which is a balanced measure in the data imbalance problem [25]. This measure lies between the range of -1 and $+1$. A value of 1 represents the perfect prediction, 0 is no better than the random guess and -1 shows total disagreement between observation and prediction.

$$MCC = \frac{cs - \mathbf{t}.\mathbf{p}}{\sqrt{s^2 - \mathbf{p}.\mathbf{p}}\sqrt{s^2 - \mathbf{t}.\mathbf{t}}} \quad (1)$$

where,

- $s = \sum_i \sum_j C_{ij}$, is the total number of samples,
- $c = \sum_k C_{kk}$, is the total number of samples correctly predicted,
- $p_k = \sum_i C_{ki}$, is the number of times class k was predicted,
- $t_k = \sum_i C_{ik}$, is the number of times class k truly occurred,

C. Implementation

We develop *StutterNet* with PyTorch library in Python [26]. We use a learning rate of 10^{-4}, amsgrad optimizer, and cross-entropy loss function. We use Librosa library [27] from Python for the feature extraction. We select models using an early stopping with a patience of seven epochs on validation loss. We compare the results obtained by *StutterNet* to existing method [20] in the same experimental framework.

V. Results

The results of our baseline *StutterNet* for different stuttering recognition are presented in Tables I and II, where we compare our technique to ResNet+BiLSTM [20]. All the considered disfluencies and the fluent speech are recognized with good scores. As can be seen from the Table I, F1-Scores show clearly the good performances of baseline *StutterNet* method in comparison to ResNet+BiLSTM. Table II also shows that the baseline *StutterNet* surpasses the state-of-the-art in most disfluency detection cases, but shows slightly lower performance in prolongation and block detection with an average accuracies of 17.13%, 42.43% in comparison to 23.17%, 53.33% average accuracies of ResNet+BiLSTM respectively. The *StutterNet* outperforms ResNet+BiLSTM in correctly detecting fluent speech with a difference of 11.63 points (66.63% for the baseline, and 55.00% for ResNet+BiLSTM).

TABLE I: Results in precision, recall and F1-score on UCLASS dataset (B: Block, F: Fluent, Rept: Repetition, Pr: Prolongation).

	Precision				Recall				F1-Score			
Method	Rept	Pr	B	F	Rept	Pr	B	F	Rept	Pr	B	F
ResNet+BiLSTM [20]	0.33	0.42	0.43	**0.63**	0.20	**0.23**	**0.53**	0.55	0.22	**0.28**	0.44	0.52
StutterNet (Baseline)	**0.36**	**0.43**	0.42	0.59	**0.28**	0.17	0.42	0.67	**0.30**	0.23	0.42	0.62
***StutterNet* (Optimized)**	0.35	0.31	**0.47**	0.59	0.24	0.13	0.47	**0.70**	0.27	0.16	**0.46**	**0.63**

TABLE II: Results in accuracies and MCC on UCLASS dataset (B: Block, F: Fluent, Rept: Repetition, Pr: Prolongation).

Method	Accuracy				Tot. Acc.	MCC.
	Rept	Pr	B	F		
Resnet+BiLSTM [20]	20.39	**23.17**	**53.33**	55.00	46.10	0.20
StutterNet *(Baseline)*	**27.88**	17.13	42.43	66.63	49.26	0.21
***StutterNet* (Optimized)**	23.98	12.96	47.14	**69.69**	**50.79**	**0.23**

We separately optimize the baseline *StutterNet* by varying the filter bank size (10, 30, 40, 50), context window (3, 7, 9, 11) and layer size (64, 128, 256, 1024). We found that context window optimization improves the detection performance of prolongation and repetition type of disfluencies. As the context increases, the performance of *StutterNet* increases for the detection of prolongation and repetition stutterings, but decreases for the fluent segments, and it remains almost unchanged for the block stuttering. This makes sense because the repetition and prolongation disfluencies usually lasts longer and the longer context helps the *StutterNet* in improving the performance. The block disfluencies doesn't last long: usually occurs at the beginning of speech segment and thus makes it context independent. We also found that layer size optimization slightly improves the performance (overall accuracy and MCC) of the stuttering detection in block and fluent types of disfluencies as shown in Fig. 2. This might be due to the possible reason that the baseline *StutterNet* is over-parameterized due to the limited size of the UCLASS dataset. We term the layer size optimized *StutterNet* as optimized *StutterNet* in Table I and II. Compared to ResNet+BiLSTM, our optimized proposed method gains a margin of 4.69% and 0.03 in overall average accuracy and MCC, respectively. For detecting the *core behaviours* and the fluent part, the margins are also substantial (improvements of 7.49%, 14.69% in repetition and fluent speech segments, respectively). Most previous work tends to avoid block disfluencies because of their similar nature to silence and prolongation (blocks are prolonged without audible airflow) [28]. As shown in Table II, the proposed *StutterNet* can detect and classify the block stuttering with an average accuracy of 47.14%. Moreover, our technique relies on the assumption that stuttering usually lasts for four-second window size [1]. Note that some of the stuttering (in particular prolongation and repetition) can exceed more than four seconds in speech [1], thus causing those prolongation stuttering likely to be misclassified.

Fig. 3 shows a visualization of the latent feature embeddings learned by *StutterNet* using t-SNE projection. Both ResNet+BiLSTM and *StutterNet* present good discrimination of the different types of disfluencies. However, the latent feature embeddings learned by *StutterNet* are more distinctive for fluent and less distinctive for prolongations and accurately capture the stutter-specific information than the state-of-the-art

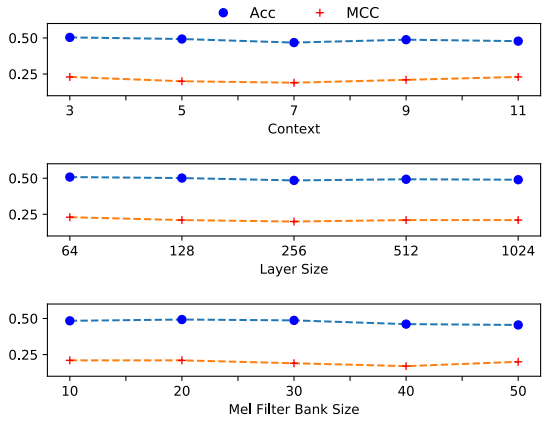


Fig. 2: MCC and accuracy of *StutterNet* with varying context, layer size and mel filter bank size (Acc is normalized in [0,1]).

ResNet+BiLSTM method. Interestingly for ResNet+BiLSTM, the fluent and repetition category's embeddings are widely spread and more overlapped with the other classes. The prolongations and blocks are well clustered in BiLSTM+ResNet as compared to *StutterNet*.

VI. Conclusion

In this work, we present a *StutterNet* to detect and classify several stuttering types. Our method uses a TDNN, which is trained on the MFCC input features. Only the *core behaviors* and fluent part of the stuttered speech were considered in this study. The results show that the *StutterNet* achieves considerable gain in overall average accuracy and MCC of 4.69% and 0.03 respectively compared to the state-of-the-art method based on residual neural network and BiLSTM. We experimentally optimize layer size, context, and filterbank size in baseline *StutterNet*. The performance moderately improves with layer size and context window optimization. Our method's main advantage is that it can detect all stuttering types with a single system with a smaller number of parameters, unlike the existing method. Our method also achieves considerable performance improvement in discriminating fluent vs. disfluent speech.

In this work, we have not evaluated the *StutterNet* on the multiple disfluencies, where two or more disfluencies are present simultaneously in an utterance. Besides, the *UCLASS* dataset was collected in a controlled environment, whereas

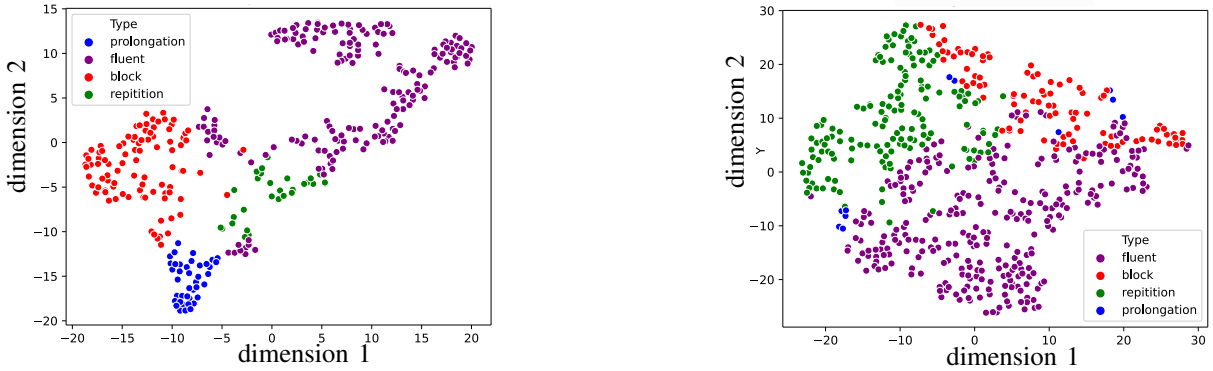


Fig. 3: t-SNE visualization of the output of last fully-connected layer for *ResNet+BiLSTM* (left) and *StutterNet* (right).

the real-time disfluency detection is a demanding problem. In future work, we will focus on multiple disfluencies by exploring the more advanced variants of TDNN for stuttering detection in a real-world scenario. We can also extend this work by exploring joint optimization of the different parameters, including context, filterbank size, and layer size of the proposed system.

Acknowledgment

This work was made with the support of the French National Research Agency, in the framework of the project ANR BENEPHIDIRE (18-CE36-0008-03). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several universities as well as other organizations(see https://www.grid5000.fr) and using the EXPLOR centre, hosted by the University of Lorraine.

References

[1] B. Guitar, *Stuttering: An Integrated Approach to its Nature and Treatment*. Lippincott Williams & Wilkins, 2013.

[2] E. Yairi and N. Ambrose, "Epidemiology of stuttering: 21st century advances," *Journal of Fluency Disorders*, vol. 38, no. 2, pp. 66–87, 2013.

[3] S. Khara *et al.*, "A comparative study of the techniques for feature extraction and classification in stuttering," in *Proc. Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 887–893.

[4] S. Alharbi *et al.*, "Detecting stuttering events in transcripts of children's speech," in *Proc. International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 217–228.

[5] S. Alharbi, M. Hasan *et al.*, "A lightly supervised approach to detect stuttering in children's speech," in *Proc. INTERSPEECH*. ISCA, 2018, pp. 3433–3437.

[6] P. A. Heeman *et al.*, "Using clinician annotations to improve automatic speech recognition of stuttered speech." in *Proc. INTERSPEECH*, 2016, pp. 2651–2655.

[7] A. B. Nassif *et al.*, "Speech recognition using deep neural networks: A systematic review," *IEEE Acess*, vol. 7, pp. 19 143–19 165, 2019.

[8] S. Latif and *at al.*, "Deep representation learning in speech processing: Challenges, recent advances, and future trends," *arXiv preprint arXiv:2001.00378*, 2020.

[9] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication*, vol. 116, pp. 56–76, 2020.

[10] L. Verde *et al.*, "Voice disorder identification by using machine learning techniques," *IEEE Access*, vol. 6, pp. 16 246–16 255, 2018.

[11] A. Waibel *et al.*, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[12] D. Snyder *et al.*, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5329–5333.

[13] P. Howell and S. Sackin, "Automatic recognition of repetitions and prolongations in stuttered speech," in *Proc. the first World Congress on Fluency Disorders*, vol. 2. University Press Nijmegen, The Netherlands, 1995, pp. 372–374.

[14] P. Howell *et al.*, "Development of a two-stage procedure for the automatic recognition of dysfluencies in the speech of children who stutter: I. psychometric procedures appropriate for selection of training material for lexical dysfluency classifiers," *Journal of Speech, Language, and Hearing Research*, vol. 40, no. 5, pp. 1073–1084, 1997.

[15] K. Ravikumar, R. Rajagopal, and H. Nagaraj, "An approach for objective assessment of stuttered speech using MFCC," in *Proc. International Congress for Global Science and Technology*, 2009, p. 19.

[16] I. Szczurowska *et al.*, "The application of kohonen and multilayer perceptron networks in the speech nonfluency analysis," *Archives of Acoustics*, vol. 31, no. 4 (S), pp. 205–210, 2014.

[17] B. Villegas *et al.*, "A novel stuttering disfluency classification system based on respiratory biosignals," in *Proc. EMBC*, 2019, pp. 4660–4663.

[18] G. Manjula, M. Shivakumar, and Y. Geetha, "Adaptive optimization based neural network for classification of stuttered speech," in *Proc. Third International Conference on Cryptography, Security and Privacy*, 2019, pp. 93–98.

[19] S. Oue, R. Marxer, and F. Rudzicz, "Automatic dysfluency detection in dysarthric speech using deep belief networks," in *Proc. of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies*, 2015, pp. 60–64.

[20] T. Kourkounakis, A. Hajavi, and A. Etemad, "Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory," in *Proc. ICASSP*, 2020, pp. 6089–6093.

[21] P. Howell, S. Davis, and J. Bartrip, "The University College London archive of stuttered speech (UCLASS)," *Journal of Speech Language and Hearing Research*, vol. 52, pp. 556–569, 2009.

[22] P. Mahesha and D. Vinod, "LP-Hillbert transform based MFCC for effective discrimination of stuttering dysfluencies," in *Proc. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 2561–2565.

[23] L. S. Chee *et al.*, "MFCC based recognition of repetitions and prolongations in stuttered speech using k-NN and LDA," in *Proc. IEEE Student Conference on Research and Development*, 2009, pp. 146–149.

[24] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations on the speaker verification task," in *Proc. SPECOM*, vol. 1, no. 2005, 2005, pp. 191–194.

[25] G. and others *et al.*, "A comparison of MCC and CEN error measures in multi-class prediction," *PloS One*, vol. 7, no. 8, p. e41882, 2012.

[26] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Advances in Neural Information Processing Systems*, 2019.

[27] B. McFee *et al.*, "librosa: Audio and music signal analysis in Python," in *Proc. the 14th Python in Science Conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[28] K. Teesson, A. Packman, and M. Onslow, "The lidcombe behavioral data language of stuttering," *Journal of Speech, Language, and Hearing Research*, vol. 46, no. 4, pp. 1009–1015, 2003.