DYSFLUENCY CLASSIFICATION IN STUTTERED SPEECH USING DEEP LEARNING FOR REAL-TIME APPLICATIONS

Melanie Jouaiti, Kerstin Dautenhahn*

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L3G5, Canada

ABSTRACT

Stuttering detection and classification are important issues in speech therapy as they could help therapists track the progression of patients' dysfluencies. This is also an important tool for technology-assisted speech therapy. In this paper, we combine MFCC and phoneme probabilities to train a neural network for stuttering detection and classification of four dysfluency types. We evaluate our system on the UCLASS, FluencyBank and SEP-28K datasets and show that our system is effective and suitable for real-time applications.

***Index Terms*—** dysfluency, stuttering, deep learning, recurrent neural network

1. INTRODUCTION

Stuttering is a neuro-developmental disorder that concerns 1% of the population. It is identifiable by core anomalies of speech such as pauses, repetition and prolongation of words or sounds and the use of interjections as filler words ("hum"). Automatic tools to detect and identify dysfluencies would be a valuable addition for therapists in order to track patients' progress but also to pave the way towards technology-assisted speech therapy.

The stuttering detection problem has been tackled from different angles with many different input features and many different methods (See [1, 2] for a review).

Mel-frequency cepstral coefficients (MFCC) are the most popular input features for stuttering detection but reported results have not been optimal or the size of the test dataset has been too small to effectively allow generalization ability which is paramount for real world applications. Besides, most of the works using MFCC only detect one type of dysfluency. For example, [3] selected 10 files from UCLASS and classified prolongation versus repetition using MFCC features with K-nearest neighbours and Linear Discriminant Analysis. [4] detected syllable repetitions using MFCC as the input feature and a combination of perceptron and dynamic time warping.

This research was undertaken, in part, thanks to funding from the Canada 150 Research Chairs Program. {mjouaiti, kerstin.dautenhahn}@uwaterloo.ca

Surprisingly, few studies employ recurrent neural networks (RNN) and prefer methods that do not take the temporal aspect of the signal into account. Using RNNs, [2] proposed a network based on ResNet and BiLSTM using audio spectrograms as the input features. They tested their network on 25 files of UCLASS and report an average accuracy of 91.15%. [5] introduced a network based on ResNet and BiLSTM and took the audio spectrogram as an input for dysfluency detection. [6] used an LSTM-based network to classify dysfluency from FluencyBank and from their dataset SEP-28K. They used a combination of mel-filterbank energy features, pitch information and articulatory features to achieve a F1-score of 83.6. They also tested phonemes' probabilities as an input feature.

Besides, other machine learning have been proposed. [7] investigated the detection of syllable repetition, word repetition and prolongation using three speech parameterization techniques as an input feature: Linear Prediction Coefficients, Linear Prediction Cepstral Coefficients and MFCC. They employed a multi-class Support Vector Machine (SVM) for classification. SVMs are also used in [8] to detect stuttering using MFCC features. [9] also explored the use of MFCC and linear predictive cepstral coefficients to detect repetitions in dysarthric speech with a deep neural network. [10] introduced StutterNet, a network based on Time Delay Neural Networks to classify dysfluency from the UCLASS datasets. They used MFCC as input and reported a total accuracy of 50.79%.

Real-time stuttering detection is also another important and overlooked aspect for the system to be usable in real-time technology-assisted therapy. For example, some works handle text transcripts [11, 12] and [5] reported excellent results, however, they trained one network for each type of dysfluency which is very impractical for real-time applications as six networks would need to run simultaneously.

In this paper, we undertake stuttering detection and dysfluency classification using a deep neural network based on BiLSTM. We use phoneme probabilities combined with MFCC as our input features and test our system on the UCLASS, FluencyBank and SEP-28K datasets.

2. METHODS

Our method extracts speech features from 3-second audio clips, applies a temporal model and outputs both a stuttering detection score and a dysfluency label for each clip. The code is written in Tensorflow (2.4.1) using Keras backend.

2.1. Datasets

In this work, we used three different datasets:

- UCLASS [13]: this dataset contains recordings from 128 children and adults who stutter. We used the annotations provided by [5]. Because [5] annotated only 25 files and did not annotate for the *block* class, we only used those files and did not use the *block* class for subsequent datasets. This dataset is extremely unbalanced with 85% of the files being fluent speech, so we oversampled the dysfluent data.
- FluencyBank [14]: this dataset contains recordings from 32 adults who stutter. We used the dataset annotated by [6] (4 144 clips) who found inaccuracies in temporal alignments for the original annotations. We removed all files where the annotations were labelled as "unsure". This dataset is mostly balanced except for *prolongations* which are under-represented.
- SEP-28K: this dataset, manually curated by [6], contains 28 177 clips extracted from publicly available podcasts. We removed all files where the annotations were labelled as "unsure".

We had approval from the University of Waterloo Ethics Board to process those datasets.

2.2. Speech features

Our input is the raw audio signal, it is subsampled to $8000Hz$ and from that we extract several speech features that will be used as inputs for the network:

- MFCC (20x47 dims) extracted with librosa [15]
- phoneme class probability (18x299 dims) and phoneme estimation (1x299 dims) extracted using [16]. The English phonetic alphabet includes 44 different phonemes, divided into 20 vowels and 24 consonants. These phonemes can be grouped into the following phonological classes considering the mode and manner of articulation: "consonantal", "back", "anterior", "open", "close", "nasal", "stop", "continuant", "lateral", "flap", "trill", "voice", "strident", "labial", "dental", "velar", "pause", "vocalic". The phoneme class probability provides the posterior probability over time for those classes and the phoneme feature provides the estimated IPA phoneme over time. Though the network was trained on Spanish languages, it has previously been used to process English language [17, 18].

We also attempted to use pitch, jitter and shimmer information as pitch often changes through dysfluency and individuals who stutter can be recognized from shimmer [19]. There were, however, no significant improvements so we will not report these results here.

2.3. Network Architecture

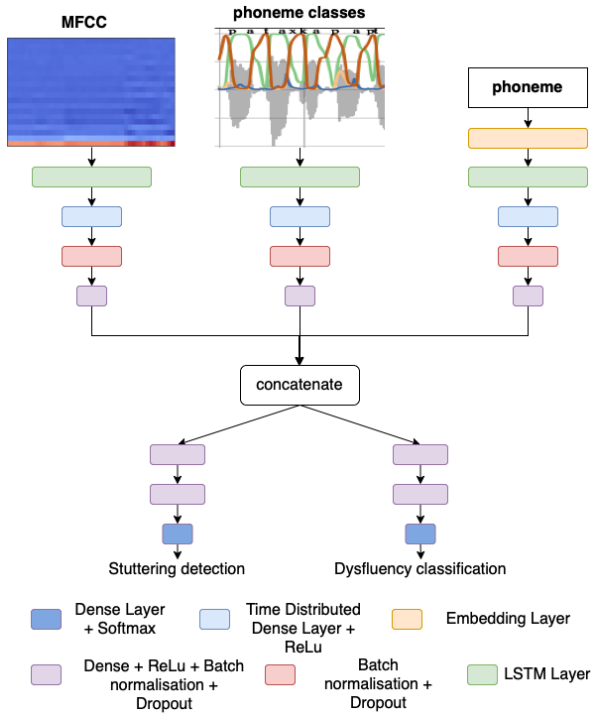


Fig. 1. Network architecture with the three input features: MFCCs, phoneme classes, phoneme

For this stuttering detection and dysfluency classification task, we propose a bidirectional LSTM (BiLSTM)-based network. LSTMs are directional models that can predict a state given the previous states. For dysfluency detection, it is useful to consider also what comes after the dysfluency and not only before so BiLSTM are a relevant option. The MFCC and phoneme class probabilities go through a BiLSTM layer with ReLU activation, then a time distributed Dense layer with ReLU activation, followed by another Dense layer. Each Dense layer is followed by batch normalisation and dropout (0.5) layers. The phoneme estimation provides string data. This data is one-hot encoded before going through an Embedding layer. After that, the architecture is the same as described previously. See Figure 2.3 for the network architecture. The networks had 128 neurons in recurrent and embedding layers.

All the outputs are combined and fed through two dense layers, each followed by batch normalisation and dropout (0.5) layers. This is followed by either a binary classification

layer (fluent or dysfluent speech) or a 5-class classification layer (fluent speech, word repetition (Wd), sound repetition (Sd), interjection (Int) or prolongation (Pro)).

2.4. Loss Functions

In therapy, it is important not to over-detect stuttering, especially when there is no stuttering as this could be detrimental for the patients. So our custom loss function for stuttering detection takes a weighted average between specificity and recall, the focus being on specificity to reduce false positives.

$$\text{custom\_loss} = 1 - (0.85 * \text{specificity} + 0.15 * \text{recall}) \quad (1)$$

For dysfluency classification, we use a custom correlation coefficient loss function. To rigorously test the proposed model, we employ 10-fold cross validation where each audio file is randomly assigned to one fold. We conducted 10 experiments where the networks learns on 90% of the data and tests on the remaining 10%. The reported results represent the average between 10 experiments. Models were trained with a batch size of 32, using the Adam optimizer, with a learning rate of 0.0001. Early stopping was used based on loss with an early stopping criteria of 15.

2.5. Evaluation & Metrics

We evaluate our model using Accuracy $\left(\frac{TP}{TP+TN}\right)$, Recall $\left(\frac{TP}{TP+FN}\right)$, F1-score $\left(\frac{TP}{TP+0.5(FP+FN)}\right)$ and specificity $\left(\frac{TN}{TN+FP}\right)$.

3. RESULTS

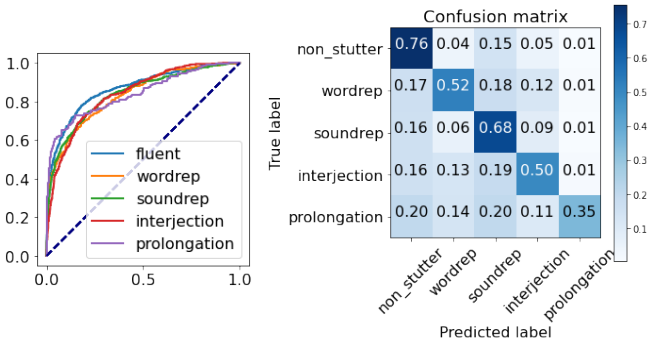
We evaluated the performance of the model with different input features: MFCC, phoneme class probabilities and both combined. We also compared our results to published results on the same datasets, when available.

3.1. Stuttering Detection

In this section, we consider the stuttering detection output of our network. For FluencyBank, our MFCCS network performed better than the other two with an accuracy of 88.3%, F1-score of 88% and recall of 82.3%. It outperformed reported results by [6] in terms of F1-score (80.8%). Regarding UCLASS, our network outperformed [10] in every aspect. However, [2] outperformed our model in terms of accuracy ([2]: 91.2%, both: 82.6%). For extensive results, see the last column of Tables 1, 2 and 3.

3.2. Dysfluency Classification

See Tables 1, 2 and 3 for the results of cross-validation on the FluencyBank and UCLASS datasets. For UCLASS, our network trained on MFCC and phoneme probabilities outperforms reported results for dysfluency classification. For FluencyBank, results with MFCC or phoneme probabilities only are slightly better than the combination of both on a case by case basis but the network trained on MFCC and phoneme probabilities still performs better overall. As an example, we provide the ROC curve and confusion matrix for the network trained on FluencyBank with both MFCC and phoneme probabilities features (See Fig. 2).



(a) ROC curve (b) Confusion matrix

Fig. 2. Results for dysfluency classification on FluencyBank

FluencyBank	Accuracy				
	Wd	Sd	Int	Pro	Any
MFCC	82.4	**79.5**	81.1	91.1	**88.3**
phonemes	**86.5**	75.3	**81.5**	**93.1**	83.9
MFCC + phonemes	86.0	78.7	81.4	93.0	82.1
UCLASS	Accuracy				
	Wd	Sd	Int	Pro	Any
Reported by [10]	24.0	NA	NA	13.0	50.8
Reported by [2]	96.6	84.1	81.4	94.1.0	**91.2**
MFCC	97.9	77.3	91.5	99.2	73.5
phonemes	97.9	86.2	89.2	99.1	71.5
MFCC + phonemes	**98.0**	**89.6**	**93.6**	**99.4**	82.6

Table 1. Accuracy results on the FluencyBank and UCLASS datasets

3.3. Generalization

Moreover, as UCLASS and FluencyBank are very small datasets, networks trained on them are unable to generalize to another dataset. To address this issue, we trained the network on different combinations of the UCLASS, FluencyBank and SEP-28K datasets (See Table 4 for the results on binary classification). In this section, the network was trained on 70% of the data, validated on 20% and tested on 10%. The data was split randomly. Regarding dysfluency classification while training on SEP-28K + FluencyBank + UCLASS, we obtained the following F1-scores: 82.9 for wordrep, 83.9 for soundrep, 82.7 for interjection and 83.8 for prolongation (See Fig. 3). When training on FluencyBank + UCLASS, we

	Accuracy	Recall	F1	Specificity
FluencyBank + UCLASS	91.0	89.4	91.1	84.7
SEP-28K + FluencyBank + UCLASS	94.3	83.8	94.1	87.6
SEP-28K + FluencyBank	96.1	80.3	95.9	86.2
SEP-28K + UCLASS	93.1	74.8	76.1	80.2

Table 4. Results on combining several datasets for binary classification

FluencyBank	F1				
	Wd	Sd	Int	Pro	Any
Reported by [6]	59.3	74.3	**82.6**	67.9	80.8
MFCC	64.1	**75.8**	74.9	47.7	**88.0**
Phonemes	**72.1**	71.3	72.4	**70.4**	83.9
MFCC + phonemes	71.5	73.8	71.5	69.8	82.2
UCLASS	F1				
	Wd	Sd	Int	Pro	Any
Reported by [10]	27			16	
MFCC	49.5	59.8	55.6	49.8	77.2
Phonemes	**70.3**	68.6	66.9	65.5	75.5
MFCC + phonemes	69.8	**72.1**	**73.2**	**70.1**	**83.9**

Table 2. F1 score results on the FluencyBank and UCLASS datasets

FluencyBank	Recall				
	Wd	Sd	Int	Pro	Any
MFCC	63.7	**78.0**	**77.4**	50.0	**82.8**
Phonemes	**70.9**	73.8	71.7	65.3	77.6
MFCC + phonemes	70.5	74.9	70.3	**69.8**	78.2
UCLASS	Recall				
	Wd	Sd	Int	Pro	Any
Reported by [10]	24			13	
MFCC	50.0	76.3	56.2	50.0	77.4
Phonemes	**69.2**	81.0	83.6	62.6	78.9
MFCC + phonemes	68.9	**81.6**	**85.3**	**63.6**	**82.2**

Table 3. Recall results on the FluencyBank and UCLASS datasets

obtained the following F1-scores: 81.1 for wordrep, 87.1 for soundrep, 86.6 for interjection and 81.5 for prolongation.

3.4. Real-time evaluation

Real-time performance is a crucial aspect for being usable in real-life settings, e.g. in technology-assisted therapeutic interventions. To evaluate time performance, we ran inference on 3s of audio data for a thousand times and took the average time. MFCC computation takes 3 ms and phoneme class inference takes 153 ms. On an i7 Intel Core CPU (11th Gen Intel Core i7-11700 @ 2.5GHz x 16), inference for the MFCC model is 18.5 ms, for the phoneme model 29.4 ms and for the model with both 30 ms. On a GeForce RTX 2060 GPU, inference for the MFCC model is 18.6 ms, for the phoneme model 49.9 ms and for the model with both 54 ms. Data processing and inference would therefore take around 200 ms which would allow audio processing at 5Hz. This should be fast enough for real-time processing of the audio data, especially since we use a $3s$ window as an input.

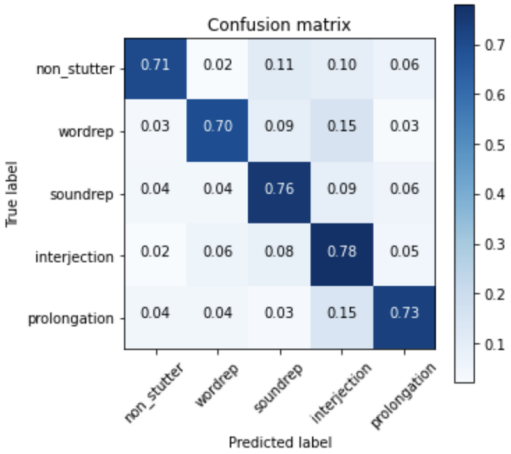


Fig. 3. Confusion Matrix for the network trained on SEP-28K + FluencyBank + UCLASS

4. CONCLUSION

In this paper we combined MFCC and phoneme probabilities to achieve stuttering detection and dysfluency classification using a BiLSTM network. Our main concerns were specificity and real-time inference so that the network can actually be usable in a real-world setting. Overall, our network performs as well or better than state of the art results, although the model trained on UCLASS notably has a lower accuracy than [2] for stuttering detection. It is however questionable whether their network would be able to achieve real time inference as six networks would need to run in parallel. Overall, performance was worse for prolongations and word repetitions. [6] reported the same issue with word repetitions and hypothesized that this was caused by their longer duration and higher variability. Regarding prolongations, this type of dysfluency is very under-represented in the UCLASS and FluencyBank datasets, which explains why the network would be struggling with them. Most limitations are due to the small size and the imbalance of the datasets UCLASS and FluencyBank. This is confirmed by the very good performance of the neural network trained on SEP-28K + UCLASS + FluencyBank or even on UCLASS + FluencyBank. Next steps involve testing the network in real-life settings to verify generalization and feasibility in technology-assisted therapy.

5. REFERENCES

[1] Shakeel Ahmad Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni, "Machine learning for stuttering identification: Review, challenges & future directions," *arXiv preprint arXiv:2107.04057*, 2021.

[2] Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad, "Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6089–6093.

[3] Lim Sin Chee, Ooi Chia Ai, M Hariharan, and Sazali Yaacob, "Mfcc based recognition of repetitions and prolongations in stuttered speech using k-nn and lda," in *2009 IEEE Student Conference on Research and Development (SCOReD)*. IEEE, 2009, pp. 146–149.

[4] KM Ravikumar, Balakrishna Reddy, R Rajagopal, and H Nagaraj, "Automatic detection of syllable repetition in read speech for objective assessment of stuttered disfluencies," *Proceedings of world academy science, engineering and technology*, vol. 36, pp. 270–273, 2008.

[5] Tedd Kourkounakis, Amirhossein Hajavi, and Ali Etemad, "Fluentnet: End-to-end detection of speech disfluency with deep learning," *arXiv preprint arXiv:2009.11394*, 2020.

[6] Colin Lea, Vikramjit Mitra, Aparna Joshi, Sachin Kajarekar, and Jeffrey P Bigham, "Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6798–6802.

[7] P Mahesha and DS Vinod, "Classification of speech dysfluencies using speech parameterization techniques and multiclass svm," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 2013, pp. 298–308.

[8] Juraj Pálfy and Jiří Pospíchal, "Recognition of repetitions using support vector machines," in *Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2011*. IEEE, 2011, pp. 1–6.

[9] Stacey Oue, Ricard Marxer, and Frank Rudzicz, "Automatic dysfluency detection in dysarthric speech using deep belief networks," in *Proceedings of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies*, 2015, pp. 60–64.

[10] Shakeel A Sheikh, Md Sahidullah, Fabrice Hirsch, and Slim Ouni, "Stutternet: Stuttering detection using time delay neural network," *arXiv preprint arXiv:2105.05599*, 2021.

[11] Kallirroi Georgila, "Using integer linear programming for detecting speech disfluencies," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, 2009, pp. 109–112.

[12] Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi, "Disfluency detection using a bidirectional lstm," *arXiv preprint arXiv:1604.03209*, 2016.

[13] Peter Howell, Stephen Davis, and Jon Bartrip, "The university college london archive of stuttered speech (uclass)," 2009.

[14] Nan Bernstein Ratner and Brian MacWhinney, "Fluency bank: A new resource for fluency research and practice," *Journal of fluency disorders*, vol. 56, pp. 69–80, 2018.

[15] Brian McFee, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, Colin Raffel, Frank Zalkow, Ayoub Malek, Dana, Kyungyun Lee, Oriol Nieto, Dan Ellis, Jack Mason, Eric Battenberg, Scott Seyfarth, Ryuichi Yamamoto, viktorandreevichmorozov, Keunwoo Choi, Josh Moore, Rachel Bittner, Shunsuke Hidaka, Ziyao Wei, nullmightybofo, Darío Hereñú, Fabian-Robert Stöter, Pius Friesch, Adam Weiss, Matt Vollrath, Taewoon Kim, and Thassilo, "librosa/librosa: 0.8.1rc2," May 2021.

[16] Juan Camilo Vásquez-Correa, Philipp Klumpp, Juan Rafael Orozco-Arroyave, and Elmar Nöth, "Phonet: A tool based on gated recurrent neural networks to extract phonological posteriors from speech.," in *INTERSPEECH*, 2019, pp. 549–553.

[17] Wenliang Dai, Samuel Cahyawijaya, Zihan Liu, and Pascale Fung, "Multimodal end-to-end sparse model for emotion recognition," *arXiv preprint arXiv:2103.09666*, 2021.

[18] Julian Fritsch and Mathew Magimai-Doss, "Utterance verification-based dysarthric speech intelligibility assessment using phonetic posterior features," *Ieee signal processing letters*, vol. 28, pp. 224–228, 2021.

[19] Kelly Dailey Hall and Ehud Yairi, "Fundamental frequency, jitter, and shimmer in preschoolers who stutter," *Journal of Speech, Language, and Hearing Research*, vol. 35, no. 5, pp. 1002–1008, 1992.