Acoustic-based fluency classification using LSTM-Attention with computationally-cheap data augmentation for an adaptive voicebot

Papa Séga Wade, Mihai Andries, Ioannis Kanellos, Thierry Moudenc

Acoustic-based fluency classification using LSTM-Attention with computationally-cheap data augmentation for an adaptive voicebot

Papa Séga WADE[1,2], *Mihai ANDRIES*[1], *Ioannis KANELLOS*[1], *Thierry MOUDENC*[2]

[1] IMT-Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
[2] Orange Innovation, France

firstname.lastname@imt-atlantique.fr, firstname.lastname@orange.com

Abstract

Most voicebots still ignore, nowadays, user fluency level, although recognizing it would allow to give answers to adaptation issues, according to the language level of the interlocutor. Towards to such an end, this paper proposes a fluency classification model using a small audio dataset. We extract various features such as Mel-frequency cepstral coefficients (MFCC) from the audio. Using recent classification models, such as CNN-LSTM-Attention and Wav2vec 2.0, we propose fluency classification models through three usual categories (Low, Intermediate and High). Furthermore, we demonstrate that simple data augmentation methods can improve classification accuracy. We employ several simple data augmentation techniques, such as speed and pitch scale variation. This augmentation multiplies by 6 the number of training samples when applied only to original samples, and by 32 when also applied to augmented samples.

Index Terms: Language fluency level, CNN-LSTM, voice conversational agent, Data augmentation

1. Introduction

Conversational agents (voicebots) are nowadays massively adopted by the general public and integrated in real-time applications used daily. The development of voicebots and vocal assistants requires automatic speech processing abilities, such as classification of users by language fluency. Yet, the automatic customer relations services do not take into account the user fluency level. Therefore, they cannot offer alternative dialogue pathways based on user language fluency level. Commonly, a single language level and/or a single interaction pattern are adopted for all customers. In order to deal with this issue, we propose a user-fluency classification method based on Deep Learning.

Our objective in this study is to estimate the user fluency level for its use within an adaptive conversational voice agent, able to adapt the linguistic complexity of its output. The target use-case is within a multi-linguistic society with a shared linguafranca (common language for inter-cultural communication), in which the population is fluent to various degrees. The socio-economic origin of the speaker is, of course, a contributing factor in language practice. It can be tracked at various levels, in terms of vocabulary, syntax, prosody, style, rhythm and especially fluency differences.

In addition, in this paper we study the influence of data augmentation in the context of fluency classification. We propose several Deep-Learning models for fluency classification, which are based on deep convolutional neural networks, LSTMs and wav2vec2 [1, 2]. The performance of the augmentation method is evaluated on the Avalinguo audio set [3]. The proposed data augmentation strategy allows to partially overcome the data challenge posed by low-resource languages.

For the evaluation of the set up models, a standard cross-validation method in 10-fold was applied. The results obtained with our models outperform the state of the art in fluency level classification [4].

The paper is structured as follows. Section 2 describes the related Work in fluency. Section 3 describes the data employed and the different data augmentation methods. The experimental models and parameters are presented in Section 3.3, followed by the results in Section 4. We conclude with a discussion on future work and perspectives.

To avoid confusion in terminology, we would like to stress the distinction between proficiency and fluency. Fluency refers to the smoothness, naturalness and flow of a speaker's speech, whereas linguistic proficiency refers to the knowledge and understanding of the grammar, syntax and vocabulary of a language. The former addresses communicability ease, the latter accuracy in using and understanding language.

2. Related Work

In recent years, there has been a growing interest in the automatic evaluation of speech fluency, particularly in the context of foreign language learning and speech therapy. In this section, we will discuss relevant studies and highlight similarities to our own use case — the analysis of speaker fluency for online voice-bot adaptation.

Regarding fluency evaluation in foreign languages, Detey et al. [5] conducted a longitudinal study on oral reading performance in French for Japanese language learners using the CLIJAF corpus [6]. The authors investigated pronunciation variations and their influence on speech fluency perception for native and non-native speakers. Their best model achieved a correlation coefficient of 0.92 between automatic and human scores.

Another recent work is that of Fu et al. (2022) [7], who investigated the use of a sequence model to learn utterance-level fluency representation from phone-level raw sequential features, using BLSTM (Bidirectional Long Short-Term Memory) and average pooling for improving non-native fluency scoring.

Additionally, Phonetic Features were used for fluency evaluation [8, 9]. The authors used phonetic fluency features to evaluate speech quality in children by employing the Forward-Backward Divergence Segmentation (FBDS) algorithm, which enabled automatic segmentation of speech signals into speech and silence segments. In addition, they predicted second language (L2) proficiency based on multi-level linguistic features.

We extend existing speech fluency evaluation methodologies (such as the Speech Rate Measurement and the Pausing Structure Analysis) to better suit an interactive voicebot. This enhancement involves classifying audio files into three fluency categories — low, intermediate, and high — without necessitating specific

annotations such as the identification of unnatural speech pauses, detection of word repetition, etc. The purpose of this adaptation is to refine the user-voicebot interaction by adjusting to different fluency levels.

3. Methods

3.1. Dataset

In this study we used the Avalinguo audio set [3]. The original dataset consists of a total of 1424 audio samples, divided into three fluency classes of non-native English speakers: Low (438 samples), Intermediate (527 samples) and High (459 samples). The files were in MP3 format sampled at 22050 Hz to 48000 Hz, which we converted to WAV format sampled at 16000 Hz with a duration of 5s each without overlapped segments.

3.2. Data augmentation

We separated the dataset into train, test and validation sets (**60% for train, 25% for test and 15% for the validation set**). We augment only the training set, keeping the test and validation sets separate to avoid information leakage. Figures 2 and 3 illustrate the use of augmentation techniques on the entire dataset.

3.2.1. ***Process 1: Perturbation methods***

Process 1 enhances the dataset's diversity through five independent perturbation methods: adding 20% white noise, altering random gain between 2 and 4, modifying pitch scale by a factor of 2 [10], applying time stretching with a factor of 0.81 [11], and randomly varying signal velocity between 0.9 and 1.1 [12]. The Python library librosa [13] was used for time stretching and pitch scaling. These perturbations were applied to the original dataset samples, increasing the training set size 6-fold, from 1424 to 8544 examples. The augmentation parameters were tested on sample audio files to ensure audio quality was maintained while simulating real-life conditions.

3.2.2. ***Process 2: Incremental composition***

For this process we managed to multiply the number of samples in our dataset by a factor of 2^5 i.e. 32. We name it *incremental composition*. We proceed first by applying random gain to modify volume or loudness of audio signal by multiplying it by a randomly generated gain factor on the wav files. We double the number of samples by applying random gain on the original audio samples. By augmenting the data set by adding 20% of white noise, time stretching, pitch scaling and random speed, we obtain 32 times more samples than in the original dataset. This gives us 45568 different audio samples divided into 3 classes: 14016 Low, 16864 Intermediate and 14688 High.

3.3. Models and experiences

When comparing our work to existing algorithmic methods for data augmentation, it is important to consider the strengths and limitations of both approaches for data augmentation: deep-learning (DL) based models, and non-DL-based models. One first limitation is the high computational cost of DL-based methods, which typically require costly GPU hardware and large quantities of data. For instance, the authors of [10] use an NVIDIA Quadro GPU to augment data using the WaveGAN approach based on the spectrogram of each WAV file. In contrast, our approach is simple and has the advantage of not needing a GPU cluster for data augmentation.

We developed four artificial neural network (ANN) models for fluency classification using three different types of data: the original data, data augmented with process 1, and data augmented with process 2. These architectures are described in the following subsections.

3.3.1. MLP

We define a MLP model using Optuna [14]. The first layer is a dense layer with 512 hidden units, l2 regularization, ReLu activation, and 50% dropout. The second layer is a dense layer with 256 hidden units, l2 regularization, ReLu activation, and 30% dropout. The third layer is a dense layer with 128 hidden units, l2 regularization, ReLu activation, and 20% dropout. The last layer is a dense layer of 3 output units with a softmax activation function. The model is trained using the Adam optimization function [15] and an exponential learning rate decay [16] with a factor of 0.6, learning rate of 0.01, and training batch size of 16 (empirically chosen).

3.3.2. CNN1D

We followed an empirical method to identify an appropriate architecture, by varying its number of layers and of neurons in each layer. Similarly, we varied hyper-parameters such as the learning rate, dropout percentage and the optimization function.

Ultimately, we chose an architecture using Optuna [14], which automatically adjusts the hyper-parameters. Optuna is used to define a *study* and a *trial* functions for each model in order to optimize the parameters and hyper-parameters for the various models. Using Optuna for architecture selection allows for a more robust and reliable model selection process, reducing the risk of overfitting, and ensuring that the chosen architecture is well-suited to the data task at hand. We customized our CNN-1D [17] architecture as follows:

- Layer 1: convolution layer of 128 filters with input (131,1), stride 1, kernel size 3, padding same, l2 regularization ($\lambda = 0.001$), batch norm, activation function ReLu and 30% Dropout, MaxPooling1D [18].
- Layers 2 and 3: convolution layers of 64 and 128 filters with the same parameters as layer 1.
- Three dense layers of 256 hidden units.
- The classification Layer: a dense layer of 3 output units with a softmax activation function to compute class probability.

The other architectures we used, such as LSTM-Attention and bi-LSTM-Attention, are all preceded by four layers of 1D convolutions.

3.3.3. LSTM- and bi-LSTM-Attention

For the LSTM-Attention and bi-LSTM-Attention models, we reused the CNN1D architecture [19] before adding these dense layers:

- Layer 1: 128 units with a batch normalization;
- Layer 2: a sequence self-attention with an activation function Tanh;
- Layer 3: 256 units with a batch normalization.

The **self-attention** layer takes as input a sequence of vectors and computes a weighted sum of the input vectors for each element in the sequence, for capturing global dependencies between input and output [20].

Given a sequence $x = (x_1, x_2, ..., x_t)$ as input, the LSTM layer produces the hidden vector $h = (h_1, h2, ..., h_t)$ and outputs $y = (y_1, y_2, ..., y_t)$ for $t \in [1, T]$ of the same length, by

iterating the following equations:

$$i_t = \sigma(W_{x_i} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f) \quad (2)$$
$$g_t = \tanh(W_{x_c} x_t + W_{hc} h_{t-1} + b_c) \quad (3)$$
$$c_t = f_t c_{t-1} + i_t \odot g_t \quad (4)$$
$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o) \quad (5)$$
$$y_t = o_t \odot \tanh(c_t) \quad (6)$$

where c_t is the state of the memory cell and i_t, f_t, o_t are gate outputs at time t [21]. The network weights W and biases b are tuned during learning to minimize the loss function. In case of a multi-layer structure the input of the next layer is the output of the previous one [21].

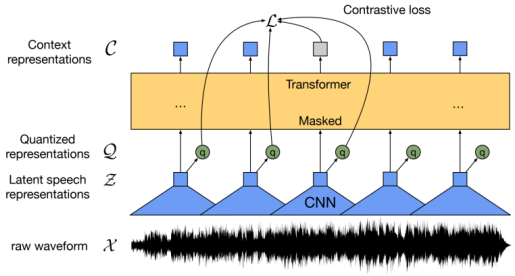

Figure 1: *Wav2vec 2.0 Architecture: learning context representations from raw waveforms [22]*

3.4. Feature extraction

3.4.1. Spectrogram

Feature extraction is a fundamental and important step for any machine learning algorithm. To perform classification on our models, we need to extract useful characteristics from the audio data. To do this, we use the librosa library [13], which provides a variety of sound features that can be extracted, including:

(i) Mel-Frequency Cepstral coefficients (MFCC) [23]. The MFCC is broken down into the following phases (P):

- P1: Split the signal into several windows that overlap each other (e.g. if we cut a signal in X windows of 256, with an overlap of 100, then the first window will be 0–255, the second 155–411, etc.). We apply the MFCC to each window.
- P2: In order to reduce spectral distortion, we apply a Hamming window to the signal [24]: $w(n) = 0.54 + 0.46cos(\frac{2\pi n}{N-1})$ where N is the length of the window. Subsequently, we multiply this function by the signal to be transformed, thus minimizing the spectral distortion created by the overlap.
- P3: Next, we apply the FFT to the window to extract the magnitude, thereby obtaining the spectrum.
- P4: We then convert to the Mel scale. Indeed, after studies on human hearing, it has been shown that humans rely on a specific frequency scale [24]. The transfer formula is simple $f_{mel} = 2595 \times log_{10}\left(1 + \frac{f}{700}\right)$ where f is the actual frequency in Hz [23].

We obtain a 128-dimensional MFCC feature vector with *n_mfcc=128* in our case.

(ii) Root Mean Square Energy (RMSE) is the square root of the mean squared amplitude over a time window [25]. It is defined by: $RMSE(x) = \sqrt{\frac{1}{N}\sum_n |s(n)|^2}$

(iii) Spectral flux: measures rate of change in spectral shape using method in [26]. This returns 1 value.

(iv) Zero-crossing-rate (ZCR): indicates the number of times that a signal crosses the horizontal axis, i.e. the number of times that the amplitude reaches 0 [27]. This feature returns 1 value. A reasonable generalization is that if the zero-crossing rate is high, the speech signal is unvoiced, while if the zero-crossing rate is low, the speech signal is voiced [28]. The zero crossing rate can be defined as follows [27]:

$$Z_n = \sum_{m=-\infty}^{\infty} \big|sng[x(m)] - sgn[x(m-1)]\big| w(n-m)$$

with

$$sng[x(n)] = \begin{cases} 1, x(n) \geq 0 \\ \text{-}1, x(n) < 0 \end{cases} \quad \text{and} \quad w(n) = \begin{cases} \frac{1}{2N} & \text{for } n \in [0, N-1] \\ 0 & \text{otherwise} \end{cases}$$

3.4.2. The pretrained wav2vec 2.0 model

To extract features from raw audio, we utilize the pre-trained model wav2vec 2.0 [22] on audio files of uniform length. Since the audio files have the same duration, zero padding was not needed to ensure consistent input size for the wav2vec 2.0 model.

Processing each audio file with the pre-trained wav2vec 2.0 model enables the extraction of relevant acoustic features. The model's training on raw, labeled audio data equips it with the ability to capture critical acoustic features suitable for diverse speech processing applications. These acoustic features are subsequently utilized in training a fluency classification model.

The model of wav2vec illustrated in Figure 1 [22] consists of a multi-layer convolutional feature encoder (also called a feature extractor) represented by the blue trapezoids. It takes as input raw audio waves X and outputs latent speech representation Z. It does this for T timesteps using a sliding window of 25 ms with a stride of 20 ms. It is pre-trained in a self-supervised setting similar to the masked language modelling used in BERT [29] for NLP. The Transformer then builds contextualized representations C over the whole input sequence X. The model is trained such that it attempts to reproduce the quantized local encoder representations in the output of the context-sensitive encoder. This training involves a procedure of randomly masking consecutive time steps within the local encoder representations. This masking process is key to challenging the model to learn to predict or "replicate" the masked portions based on the surrounding context.

To evaluate the prediction quality of our model, we use the objective function L_m defined bellow. This function L_m is used to predict audio features from context or to predict missing features during the masked language modeling task.

$$L_m = -\log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \in Q_t} \exp(\text{sim}(c_t, \tilde{q})/\kappa)}$$

where c_t represents the context vector for a given audio segment, q_t represents the quantized target vector for the masked audio feature, $\tilde{q}$ represents a quantized vector in the vocabulary Q_t, and $K = 100$ distractors, κ is the temperature parameter which is set to 0.1. The similarity function $\text{sim}(\cdot, \cdot)$ is typically [30] the cosine similarity function $\text{sim}(u, v) = \frac{u^\top v}{\|u\| \|v\|}$.

	MLP			CNN1D			LSTM-Attention			biLSTM-Attention		
	Precision	Recall	**F1-score**	Precision	Recall	**F1-score**	Precision	Recall	**F1-score**	Precision	Recall	**F1-score**
Low	86.29	93.21	89.61	94.81	90.12	92.41	93.98	93.28	**93.63**	94.29	90.83	92.52
Intermediate	97.87	97.87	**97.87**	99.25	94.33	**96.73**	99.13	96.98	**98.13**	95.88	97.89	**96.88**
High	91.96	82.40	86.92	85.71	96.00	90.57	91.67	96.12	**93.84**	90.36	92.59	91.46

Table 1: *Classification scores for each evaluated model (in percentages)*

Table 2: *Classifier test scores, per model*

Metrics	MLP	CNN1D	LSTM-Attention	biLSTM-Attention
Accuracy	91.59	93.22	**95.98**	94.39
Precision	91.81	93.17	95.32	93.75
Recall	91.59	93.12	94.98	93.89
F1-Score	91.69	93.14	**95.15**	93.81

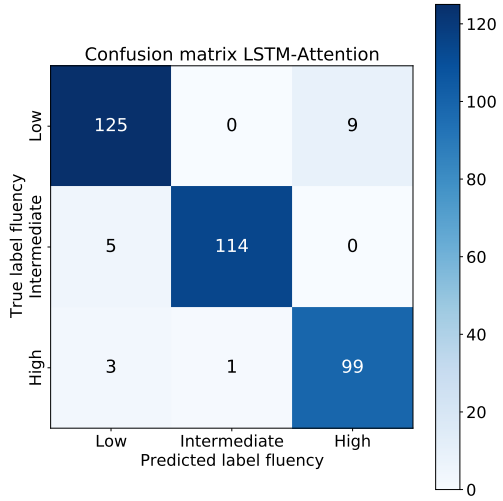

Figure 2: *Confusion matrix LSTM-Attention model*

4. Results and Discussion

To evaluate our results across the different models (see Table 2), we used the accuracy and F1-score [31]. The accuracy metric measures the ratio of correct predictions over the total number of evaluated instances [32]. The F1-Score is defined from the Precision and the Recall. Precision is defined as the ratio between the number of true positives and the total number of positives predicted by the model [32]. The Recall is defined as the fraction of positive samples that are correctly classified [32]. The F1-score is the harmonic mean of precision and recall, and is widely used as a measure of classifier performance [33].

We evaluated the classification scores of each model (see results in Table 1). To evaluate our models we performed a 10-fold cross validation [34] with a shuffling of the data explained.

Table 3: *Results obtained using 10-fold cross-validation*

Metrics (means)	MLP	CNN1D	LSTM-Attention	biLSTM-Attention
Accuracy	93.62	94.82	95.97	93.07
Precision	94.83	94.91	95.76	93.47
Recall	94.82	94.81	95.85	92.97
F1-Score	94.82	94.86	**95.80**	93.21

The results are presented in Table 3. We would like to remind that the distribution of data by class is not exactly balanced: 30.76% Low, 37.01% Intermediate, 32.23% High.

Our experiment demonstrated that our proposed method is efficient at accurately classifying audio samples into the correct categories of speaker fluency level. We used a dataset composed of 1424 samples for our experiment, of which 60% were used for training, 15% for choosing the hyper-parameters, and 25% for testing the performance of our model. Our model achieved an overall accuracy on the test set of 95.44% on the original dataset and 96.21% on the augmented dataset. Furthermore, we performed a detailed analysis of the model's performance, and found that it achieved a high level of accuracy for all of the three categories in Table 1. The LSTM-Attention model shows slightly higher F1 score for each class. The *Intermediate* category had the highest F1-score at 98%, while the *Low* and *High* categories had the same F1-score of 93%.

There are a few potential reasons for the relatively lower accuracy on the *Low* and *High* categories:

(i) the *Low* category is inherently more difficult to classify, as it can include a poor representation in word content in the signal

(ii) unbalanced classes in the dataset, with more samples of *Intermediate* category than others.

We also utilized the Wav2vec 2.0 [22] model with the SpeechBrain toolkit [35] mentioned in the feature extraction section for the classification by using the IEMOCAP recipe [2] in our experiments. The Wav2vec 2.0 model designed to learn high-level speech representations directly from raw audio waveforms, without requiring any manual feature engineering. Wav2vec 2.0 was able to capture important features of the audio samples in our dataset with promising results (95.23% accuracy on the test set).

5. Conclusion

We proposed an implemented data augmentation technique to enhance the robustness and generalization ability of our models for fluency classification. Specifically, we generated additional training samples by applying transformations to the original samples, such as adding white noise, pitch scaling, time stretching, random gain, and signal speeding. By doing so, we were able to increase the number of training examples by a factor of 32.

We compared the performance of MLP, CNN1D, LSTM-Attention and biLSTM-Attention for classifying users based on the fluency of their speech. We determined that the best performing model on the test set was a Long Short Term Memory (LSTM) model with Attention. It was trained on the augmented dataset using features such as MFCCs. The overall accuracy was 95.4% on the original dataset and 96.2% on the augmented dataset.

Future studies could explore other information channels to obtain a more through evaluation of user's fluency. It may be useful to measure speech rate in words per minute, assess vocabulary, monitor unnatural pauses or hesitations from the transcription. We intend to expand our research on these ideas.

6. References

[1] L. Goncalves and C. Busso, "Improving Speech Emotion Recognition Using Self-Supervised Learning with Domain-Specific Audiovisual Tasks," in *Proc. Interspeech 2022*, 2022, pp. 1168–1172.

[2] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *ArXiv*, vol. abs/2111.02735, 2021.

[3] A. P. Grijalva, "Avalinguo audio set," https://github.com/agrija9/Avalinguo-Audio-Set.

[4] J. A. V. Mora, "Identification of pronunciation errors in l2 english speech by spanish speaking natives for s-impure sounds," 2020.

[5] S. Detey, L. Fontan, M. Le Coz, and S. Jmel, "Computer-assisted assessment of phonetic fluency in a second language: a longitudinal study of japanese learners of french," *Speech Communication*, vol. 125, pp. 69–79, 2020.

[6] V. De Fino, L. Fontan, S. Detey, I. Ferrané, and J. Pinquier, "Corpus de parole non-native et prédiction automatique du niveau de performance en expression orale : application à CLIJAF," in *Journées Interphonologie du Français Contemporain (IPFC 2022)*, Paris, France, Dec. 2022, pFC (Phonologie du Français Contemporain) est un programme de recherche offrant une base de données de français oral contemporain dans l'espace francophone. [Online]. Available: https://hal.science/hal-03946408

[7] K. Fu, S. Gao, X. Tian, W. Li, and M. Zejun, "Using Fluency Representation Learned from Sequential Raw Features for Improving Non-native Fluency Scoring," in *Proc. Interspeech 2022*, 2022, pp. 4337–4341.

[8] L. Fontan, S. Kim, V. De Fino, and S. Detey, "Predicting speech fluency in children using automatic acoustic features," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 1085–1090.

[9] V. De Fino, L. Fontan, J. Pinquier, I. Ferrané, and S. Detey, "Prediction of L2 speech proficiency based on multi-level linguistic features," in *Proc. Interspeech 2022*, 2022, pp. 4043–4047.

[10] A. Madhu and S. Kumaraswamy, "Data augmentation using generative adversarial network for environmental sound classification," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[11] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[12] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[13] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[14] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] K. You, M. Long, J. Wang, and M. I. Jordan, "How does learning rate decay help modern neural networks?" *arXiv preprint arXiv:1908.01878*, 2019.

[17] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.

[18] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d & 2d cnn lstm networks," *Biomedical signal processing and control*, vol. 47, pp. 312–323, 2019.

[19] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] I. Lezhenin, N. Bogach, and E. Pyshkin, "Urban sound classification using long short-term memory neural network," in *2019 federated conference on computer science and information systems (FedCSIS)*. IEEE, 2019, pp. 57–60.

[22] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[23] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[24] M. Sahidullah and G. Saha, "On the use of distributed dct in speaker identification," in *2009 Annual IEEE India Conference*. IEEE, 2009, pp. 1–4.

[25] M. B. Er, "A novel approach for classification of speech emotions based on deep and acoustic features," *IEEE Access*, vol. 8, pp. 221 640–221 653, 2020.

[26] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[27] R. G. Bachu, S. Kopparthi, B. Adapa, and B. D. Barkana, "Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy," in *Advanced techniques in computing sciences and software engineering*. Springer, 2010, pp. 279–282.

[28] L. R. Rabiner, *Digital processing of speech signals*. Pearson Education India, 1978.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *ArXiv*, vol. abs/2002.05709, 2020.

[31] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.

[32] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[33] G. Forman *et al.*, "An extensive empirical study of feature selection metrics for text classification." *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1289–1305, 2003.

[34] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.

[35] T. Parcollet, M. Ravanelli, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. de Mori, and Y. Bengio, "SpeechBrain: A General-Purpose Speech Toolkit," Mar. 2022, preprint. [Online]. Available: https://hal.science/hal-03601303