MLADDC: Multi-Lingual Audio Deepfake Detection Corpus

Arth J. Shah, Ravindrakumar M. Purohit, Dharmendra H. Vaghera, Hemant A. Patil

Speech Research Lab
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar (India)
{202101154, 202321002, 202311054, hemant_patil}@daiict.ac.in

Abstract

This study develop Multi-Lingual Audio Deepfake Detection Corpus (MLADDC) to boost the ADD research. Existing datasets suffer from several limitations, in particular, they are limited to one or two languages. Proposed dataset contains 20 languages, which have been released in 4 Tracks (6 - Indian languages, 14 - International languages, 20 languages half-truth data, and combined data). Moreover, the proposed dataset has 4×10^5 files (1,125+ hours) of data, which makes it one of the largest datasets. Deepfakes in MLADDC have been produced using advanced DL methods, such as HiFiGAN and BigVGAN. Another novelty lies in its sub-dataset, that has partial deepfakes (*Half-Truth*). We compared our dataset with various existing datasets, using cross-database method. For comparison, we also proposed baseline accuracy of *68.44%*, and EER of *40.9%* with MFCC features and CNN classifier (14 languages track only) indicating technological challenges associated with ADD task on proposed dataset.

1 Introduction

Deepfakes are artificially generated fake media using deep learning (DL) methods. Recent study found that deepfakes are challenging to detect even for human listeners, however, machines can do better job in their detection [1]. Audio Deepfake Detection (ADD) system needs a statistically meaningful dataset to be able to train a reliable model. There exist several datasets for ADD task (to be discussed in sub-Section 1.1), however, they suffer with several limitations. One of the key limitations is number of languages used in dataset, i.e., most of the datasets are restricted to a single language (English), or a few number of languages. Thus, multi-lingual dataset is needed to obtain an generalized model for ADD. Our proposed dataset has 4×10^5 files and has 20 languages, which have been released in 4 tracks (sub-dataset {*Indian Languages - T1*}, super-dataset {*International Languages - T2*}, half-truth generated audio files {*20 Languages - T3*}, and {*combined dataset - T4 (T1 + T2 + T3)*}). We employed Generative Adversarial Networks (GANs) for deepfake generation because unlike conventional neural networks that are trained on supervised tasks (such as classification or regression), GANs are unsupervised, and specialize in learning to generate new data. They involve a generator and a discriminator competing against each other, which is different from a single network architectures. On the other hand, while autoencoders compress and reconstruct input data, GANs generate completely new data. In audio, Variational Autoencoders (VAEs) are often used for tasks, such as speech generation, but GANs can produce sharper and more realistic outputs, although they can be more difficult to train [2]. Recurrent models, such as RNNs or LSTMs are typically used for sequential tasks, such as audio classification or speech recognition. GANs, on the other hand, focus more on data generation and style transfer rather than sequential prediction. WaveNet models are typically used for high quality audio synthesis and are based on autoregressive models. GANs comparatively offer a faster generation process as they do not require sequential processing. To our best knowledge and belief, this is the first study of its kind that proposes the corpus, which contains both the real and the corresponding fake utterances of each speaker *w.r.t.* the same text material used for the recordings. This study offers the following novelty :

- HiFi-GAN and BigVGAN are used for deepfake audio generation,

- Multi-lingual deepfake audio generation,
- Multi-lingual *half-truth* audio generation,
- Semi-supervised learning for fake audio generation.

1.1 Related Work

Previously a few datasets have been proposed, which also have been released in various tracks. The most popular among them is Fake or Real (FoR) dataset [3], which has been released in 4 parts, and has around 69K files. Deepfakes in FoR dataset have been generated by 7 different types of TTS models, and has a total of 33 types of different speakers. Another interesting dataset has been proposed in ADD 2022 [4] and ADD 2023 challenge [5], which generated deepfakes from various unknown algorithms. WaveFake dataset uses MelGAN in order to generate deepfakes from Mel spectrogram of raw audio [6]. In-the-wild consist of around 30K files, generated from 19 different types of TTS models, which is also restricted to English language only. In [7], authors proposed a multi-lingual dataset, however, it is restricted to spoofing tasks only as audio are generated via TTS, and audio were unable to maintain speaker / language-specific characteristics. MLAAD generated spoofed audio using 54 TTS models and 21 different architectures. Other datasets, such as Half-Truth Audio Detection (HAD) dataset [8], employ LPCNet vocoder to generate deepfake and provides around 160K files. On the other hand, GANs have been used for various speech-based applications, and also for image-based applications [9, 10, 11].

2 Proposed Methodology

Study reported in [12] proposed WaveNET; the core autoregressive architecture of WaveNET was obtained by deleted convolutional layers, where the current audio sample is sequentially conditioned based on the previous samples to predict the probability distribution of the current one to capture the complex dependencies of the prior sample of waveform. The limitation of this autoregressive architecture was that it could not grasp patterns of future audio samples. Real-time synthesis was challenging since it generated each audio sample sequentially, although optimization and parallelization techniques have been developed. This gap created inconsistencies and unnatural artifacts in the generated speech. To minimize these limits, flow-based speech synthesizers are used as the teacher network to train the student network, where student network use the maximum likelihood to reduce the difficulty of the training model [13]. Also, flow-based models can capture complicated long-term dependencies, but again, flow-based models take enormous computational resources compared to autoregressive architectures, where a few parameters can be trained with a constrained number of resources. GAN-based speech synthesizers [14, 15, 16, 17, 18, 19, 20] solves the issue of the stability of the large scale training, and training inference speed over the previous two architectures. GAN consists of two neural networks, namely, the Generator (G) and the Discriminator (D), that are trained together in a competitive process. G creates synthetic data (such as images, videos, or audio) from random noise, while the D tries to distinguish between real data and the data produced by the G. As they train, the G gets better at creating realistic data, and the D improves at identifying fake data. Eventually, the G becomes so skilled that the fake data is nearly indistinguishable from real data. In [21], authors proposed WaveGAN, which relies on the intermediate representation of Mel spectrogram, but the generated speech was a very low level of the speech as compared to the state-of-the-art methods and also faced issues with stability due to the early GANs applications. MelGAN [17] enhanced the quality of synthesized speech by generating more natural and lifelike sounds, building upon work of WaveNet [12]. However, the production of artifacts in the high frequency samples leading to compresion. Study in [20] proposed Parallel WaveGAN, which supports parallel synthesis, such as flow-based models, and Kong *et al.* [18] proposed the HiFiGAN, which provides high quality and fast inference by reducing the artifacts issue in the generated speech samples, thereby motivating us to use HiFi-GAN for generating deepfake signals. Also, in some cases, both architectures cannot perform well for the unseen data. Hence, study in [19] proposed BigVGAN to focus on the scaling the model and generate the diverse speech output for the various conditions. Further, BigVGAN successfully captures and handles the diverse range of voice styles and languages with minimal fine-tuning and pushes the boundary of speech synthesis by setting its standards, which makes it a best approximation to generate deepfakes. GANs are ideal for deepfake generation because they excel in producing highly realistic synthetic data. Their ability to learn complex distributions from real-world data allow them to generate convincing, high quality deepfakes that are difficult to differentiate from authentic (real) audio.

2.1 HiFi-GAN

We employed HiFiGAN for generating deepfake due to its ability to produce high quality and high-fidelity audio. It utilizes two discriminators: **(1) Multi-Period Discriminator (MPD), and (2)**

Multi-Scale Discriminator (MSD) at different temporal resolutions, ensuring that the generated waveform is both perceptually convincing and closely aligned with real-world data. For multilingual deepfake generation, efficiency of HiFi-GAN in synthesizing clear and natural-sounding speech across various languages makes it well-suited for applications requiring fast and real-time generation. The model can generalize the diverse linguistic sounds while maintaining clarity, making it advantageous for deepfake involving speech in multiple languages.

The major advantage of employing HiFi-GAN is its speed of inference without sacrificing quality. It is designed for efficient generation, which allows for real-time vocoding, crucial in practical deepfake systems, where performance is the key. Additionally, it maintains low computational cost compared to the traditional GANs, offering a balance between quality and speed, which is valuable when working with multiple languages and large datasets.

2.2 BigVGAN

The autoregressive-based speech synthesizer produces the natural speech, one sample at a time [22, 23]. In the real-time scenario, it is very slow due to the sequential generation of the samples. However, this sequential nature is also less scalable for long-term speech generation. In such cases, the artifacts are produced during the inference due to the limited capabilities of latent space exploration. Overcoming this flow-based synthesizer comes with parallel processing in training and inference. Which increased the scalability and control over the input data distribution. It has become very complex in large-scale training because of the sequence of invertible transformations (flows) that map data to a latent space and back. As model scales, the architecture of the flow layers and sensitivity to these hyperparameters can increase, making training more complex and time-consuming.

Built on the strengths of HiFi-GAN by scaling up its architecture, making it even more suitable for generating deepfake audio files in a variety of languages. BigVGAN achieves higher fidelity than its' HiFiGAN counterpart by incorporating a more robust and flexible architecture that allows for better handling of complex audio features. This results in superior audio quality, especially for tasks involving nuanced sounds, emotions, and intricate speech patterns across multiple languages. The larger model capacity enables BigVGAN to deliver state-of-the-art performance for deepfakes, ensuring more realistic and coherent results even for difficult-to-synthesize languages.

One of BigVGAN's primary advantages is its improved generalization, meaning it can handle unseen data and new languages more effectively. This makes it ideal for multi-lingual deepfake generation, where the diversity of languages might pose challenges. Its use of advanced training techniques helps ensure that the model doesn't overfit on specific language characteristics and can adapt to the varied structures and sounds of different languages. The high-fidelity output it provides can be particularly valuable for applications requiring premium quality deepfake audio.

2.3 Data Generation

We employed the HiFiGAN [18] and BigVGAN [19] pre-trained models (PTE), $\theta_{HiFi-GAN}$, $\theta_{BigVGAN}$, which are available publicly in order to generate deepfakes. Both the model were selected after examining their ability to generate deepfakes. HiFiGAN and BigVGAN were trained on VCTK[24] and LibriTTS[25] corpus with 14M and 112M parameters, respectively. As DL models focus more on shape of signal rather than amplitude of signal, and generalization of model over unseen data, the issue of volume normalization was observed on deepfakes, which was further normalized via similar method employed in [8]. In Algorithm 1, X_{Data} represent the dataset (combination of the train, test, and valid sets), $V.Norm.$ represents normalization of volume w.r.t. the original files to ensure consistency and naturalness of deepfake audio. X_D serves as the input to the PTE model, while $\hat{Y}_D$ denotes the corresponding output obtained from the model. The weight normalization process is denoted by W_{norm}, which ensures that the model parameters are appropriately chosen throughout the process.

3 Details of MLADCC

This Section presents details of proposed dataset structure and its design. Due to the limited language resources, we were unable to collect real audio samples data manually. Alternatively, in this study, we propose a dataset in which we generated 160k deepfake samples of 80k real utterances, which were collected from the VoxLingua107 dataset [26], which is one of the most popular and largest open source multilingual dataset for Spoken Language Identification (SLID) task. VoxLingua107 was formed by recording utterances from 107 different languages and data from 6628 hours. Limited to storage resources, authors could not create a dataset for more than 20 languages, namely, Sanskrit, Hindi, Bengali, Tamil, Gujarati, Punjabi, Arabic, Mandarin Chinese, English, French, Finnish, German, Indonesian, Japanese, Portuguese, Russian, Spanish, Swedish, Urdu, and Vietnamese.

Algorithm 1 Inference with PTE, HiFiGAN, and BigVGAN.

X_{Data} V. Norm. $RawData_{(Train,Test,Valid)}$
Device $\leftarrow \theta_{HiFi-GAN}(PTE_{HiFi-GAN})$,
Device $\leftarrow \theta_{BigVGAN}(PTE_{BigVGAN})$,
for each $X_D \in \{\mathrm{X}_{Data}\}$ **do**
    Device $\leftarrow (X_D)$
    $\hat{Y}_D[]input/PTEf_\theta(X_D)$
    $Save(\hat{Y}_D, X_D)$
end for
$\theta'_{HiFi-GAN}, \theta'_{BigVGAN} \leftarrow \mathrm{W}_{norm}(\theta_{HiFi-GAN}, \theta_{BigVGAN})$,
for each $Data \in \{training, testing, validation\}$ **do**
    $G_{output}\ \hat{Y}_D \leftarrow f_{\theta'}(X_D)$
    $Path\{(\hat{Y}_{Training}, \hat{Y}_{Test}, \hat{Y}_{Valid}\} \leftarrow Save(\mathrm{G}_{output})$
end for

Average of 11.35 hours of data was selected from each language on basis of time duration statistics. Comprising 20 languages in the MLADDC, it is also robust to dialects. The total number of utterances in the proposed dataset is 4×10^5 files (8×10^4 real, 16×10^4 deepfake, and 16×10^4 half-truth), making it one of the largest datasets among currently available open source datasets in the ADD literature. Dataset statistics and demo is publicly available at [1].

3.1 Real Data

First, we collected all the real audio samples available from the VoxLingua107 dataset (open source and freely available) [26]. We labeled them into 5 classes based on the audio duration of particular samples, namely, A (0-5 seconds), B (5-10 seconds), C (10-15 seconds), D (15-20 seconds), and E ($>$ 20 seconds). After that, we selected 1,000 random samples (except in Sanskrit class) from each class collectively to form a dataset of total of 225.13 hours (80,000 audio samples) of real data. We eliminated the issue due to audio sample size dependencies by selecting the variable length audio. In order to generalize sampling rate to *16 kHz*, all audio of VoxLingua107 were resampled to *16 kHz* before generating deepfake from it. The resampling process was carried out in order to generalize the dataset.

3.2 Fake Data

We use the model based on HiFi-GANs and BigVGANs to generate 16×10^4 (8×10^4 for each) deepfakes from the real signal (described in sub-Section 2.1 and 2.2), which resulted into total 450.26 hours of deepfake data. We employed to process the real audio and generate the deepfake audio of the same speaker with the same utterance spoken in the original samples. Both HiFi-GAN and BigVGAN generated deepfake illustrate properties similar to those of real signals. Due to their perfect generation (i.e., high *perceptual* similarity), these generated deepfakes are extremely difficult to distinguish by human listners. As the dataset is generated by sophisticated ML / DL methods, it also aims to fool the humans as well as ADD system.

3.3 Half-Truth Data

We generated total of 16×10^4 partially fake files (i.e., Half-truth), out of which 8×10^4 fake audio were generated using BigVGAN, and another 8×10^4 audio generated from HiFi-GAN. For generating Half-Truth audio, we selected real audio from each languages and then, mapped the corresponding deepfake generated via HiFi-GAN. We replaced around one second of real audio with deepfakes (once HiFi, and then BigV), which resulted into total 450.26 hours of half-truth data. Time of replacement was chosen randomly, and data statistics were noted. We did not replace a particular word from audio signal, rather replace an random portion of signal, which may be even an half word, because if we replace only word and not an random phase of speech, the systems based on tokenization can easily tokenize the sentence into words, and detect deepfake words easily. On the other hand, if the word is half fake and half true, we believe that even the models trained based on tokenization will not be able to detect the difference between deepfake *vs.* real. More mathematics and detailed analysis on half-truth can be found on [27].

4 Experimental Results

4.1 Baseline Results

Experiments are performed using two baseline features, i.e., MFCC, and LFCC using existing well known pattern classifiers, such as BiLSTM, CNN, BiGRU, and ResNet-50. Details and codes related

[1]https://speech007.github.io/MLADDC_Nips/

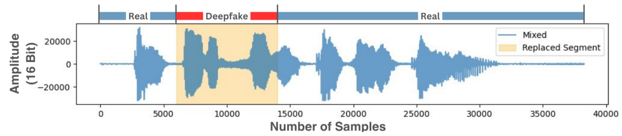

Figure 1: Illustration of generation of half-truth data.

to features employed and classifiers used can be found on Appendix A. Results indicate very large EER ($\approx 50\%$) and around, so most of the audio files were classified as deepfake (i.e., more false alarm) due to bias training of model on *MLADDC* dataset. Almost every audio was predicted as deepfake and only a few real audio were predicted correctly. It can be observed from Table 1, when we move on to critically generated dataset, i.e., from T1 to T2, the number of language increases, and accuracy drop can be observed due to increase in complexity of dataset. Moreover, it can also be observed that not even skip connection-based model (ResNet-50) is able to detect generated deepfakes, proving the superiority of crucially generated deepfakes.

Table 1: Comparison of Results on three trackes T1, T2, and T3 of MLADDC (C* ->Classifiers, TA ->Testing Accuracy, EER ->Equal Error Rate).

Feat.	C*	T1		T2		T3	
		TA	EER	TA	EER	TA	P
LFCC	{1}	68	43.7	67.36	47.9	58.01	58.78
	{2}	66.66	50	68.44	40.9	56.94	56.87
	{3}	66.29	48.9	67.06	48.91	57.54	58.28
	{4}	44.88	50	33.39	50.9	47.14	57.05
MFCC	{1}	73.43	32.6	66.66	50	56.86	56.89
	{2}	69.16	41.5	66.66	50	56.86	56.89
	{3}	68.86	42.4	66.66	50	56.86	56.89
	{4}	51.81	50.2	66.66	50	56.86	56.89
{1} ->BiLSTM		{2} ->CNN		{3} ->BiGRU		{4} ->ResNet-50	

4.2 Cross-Database Evaluation

We observed cross-dataset evaluation on a few of the existing dataset, inorder to prove superiority of the dataset proposed. For this task, we examined results on three popular open-source deepfake datasets, namely, FoR [3], In-The-Wild [28], and ASVSpoof [29]. Not every dataset is an open-source dataset, which is another limitation for performing cross-database evaluation in this study. Table 2 denoted the accuracies obtained when the existing datasets are self-testing (training and testing on existing data), and MLADDC Testing (training on existing dataset, and testing on MLADDC). Results of cross training (training on MLADDC, and testing on other datasets) can be found on Appendix B.

Table 2: Results (Accuracy in %) on Cross-Database Scenario using MFCC as feature and BiLSTM as classifier.

Fetures	Train dataset	Self Testing	MLADDC Testing (T2)
MFCC	FoR	65.75	37.31
	ITW	66.67	33.33
	ASVSpoof	91.98	33.84
LFCC	FoR	84.61	56.23
	ITW	99.02	59.19
	ASVSpoof	95.6	33.34

5 Summary and Conclusions

This study proposed an novel multi-lingual dataset, in which deepfakes are generated by using HiFiGAN, and BigVGAN. It also includes half-truth audio. Proposed dataset is one of the largest dataset for ADD, as well as HAD tasks, with an total duration of 1125+ hours and 4×10^5 files in total. We also conducted baseline experiments in order to evaluate efficiency of dataset. In order to prove superiority of proposed dataset, we also trained model on various existing datasets, and tested on proposed dataset. We in future plan to release dataset challenge, for both deepfake detection and half-truth detection. Current limitations of study include training of HiFiGAN, and BigVGAN, which has been done on LJspeech, and VCTK corpus, which are only English language. Our future plan is to retrain GANs modes on multi-lingual dataset to generate more realistic deepfakes, thereby resulting into an open research challenge. Additionally, we plan to expand our approach by incorporating a range of classifiers, specifically Transformer-based BERT and XLNet. These models, with advanced attention mechanisms, are suitable to handle lengthy sequences, which is essential for deepfake detection. This will allow in-depth analysis of multilingual phonetic classification and temporal anomalies for deepfake detection across multilingual.

References

[1] Kimberly T Mai et al. “Warning: Humans cannot reliably detect speech deepfakes”. In: *PLOS One* 18 (2023), pp. 1–20.

[2] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems (NIPS)* 27 (2014). Montréal, Canada, 2672—2680.

[3] Ricardo Reimao and Vassilios Tzerpos. “FoR: A dataset for synthetic speech detection”. In: *International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. 2019, Timișoara, Romania, pp. 1–10.

[4] Jiangyan Yi et al. “ADD 2022: The first audio deep synthesis detection challenge”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, Singapore, pp. 9216–9220.

[5] Jiangyan Yi et al. “ADD 2023: The Second Audio Deepfake Detection Challenge”. In: *arXiv preprint arXiv:2305.13774* (2023). {Last Accessed Date: 16$^{\text{th}}$ August, 2024}.

[6] Joel Frank and Lea Schönherr. “WaveFake: A DataSet to Facilitate Audio Deepfake Detection”. In: *arXiv preprint arXiv:2111.02813* (2021). {Last Accessed Date: $16^{th} August, 2024$}.

[7] Nicolas M Müller et al. “MLAAD: The Multi-Language Audio Anti-Spoofing Dataset”. In: *arXiv e-prints* (2024). {Last Accessed Date: 16$^{\text{th}}$ August, 2024}, arXiv–2401.

[8] Jiangyan Yi et al. “Half-Truth: A Partially Fake Audio Detection Dataset”. In: *INTERSPEECH*. Brno, Czechia. 2021, pp. 1654–1658.

[9] Hajar Emami et al. “SPA-GAN: Spatial attention GAN for image-to-image translation”. In: *IEEE Transactions on Multimedia* 23 (2020), pp. 391–401.

[10] Diego Gragnaniello et al. “Are GAN generated images easy to detect? A critical analysis of the state-of-the-art”. In: *IEEE international Conference on Multimedia and Expo (ICME)*. 2021, virtual, pp. 1–6.

[11] Nagaraj Adiga et al. “Speech Enhancement for Noise-Robust Speech Synthesis Using Wasserstein GAN”. In: 2019, Graz, Austria, pp. 1821–1825.

[12] Aaron van den Oord et al. “WaveNet: A Generative Model for Raw Audio”. In: *9th ISCA Speech Synthesis Workshop (SSW9)*. Sunnyvale, California, USA, 2016.

[13] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. “WaveGlow: A Flow-based Generative Network for Speech Synthesis”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom, 2019, pp. 3617–3621.

[14] Zhifeng Kong, Wei Ping, and Bryan Catanzaro. “Glow-WaveGAN: Learning Speech Representations from GAN-based Variational AutoEncoder for High-Fidelity Flow-Based Speech Synthesis”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, Canada (Virtual Conference), pp. 6024–6028.

[15] Zhehuai Chen et al. “Improving Speech Recognition Using GAN-Based Speech Synthesis and Contrastive Unspoken Text Selection”. In: *INTERSPEECH*. Shanghai, China (Virtual Conference), 2020, pp. 556–560.

[16] Xin Wang, Shinji Takaki, and Junichi Yamagishi. “Wasserstein GAN and Waveform Loss-based Acoustic Model Training for Multi-speaker Text-to-Speech Synthesis Systems Using a WaveNet Vocoder”. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, UK, 2019, pp. 6830–6834.

[17] Kundan Kumar et al. “MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis”. In: *Advances in Neural Information Processing Systems (NIPS)* 32 (2019), pp. 14881–14892.

[18] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis”. In: *Advances in Neural Information Processing Systems (NIPS)* 33 (2020). Vancouver, Canada, pp. 17022–17033.

[19] Sang gil Lee et al. “BigVGAN: A Universal Neural Vocoder with Large Scale Training”. In: *The Eleventh International Conference on Learning Representations (ICLR)*. 2023, Kigali, Rwanda.

[20] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. “Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram”. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, Barcelona (virtual conference), pp. 6199–6203.

[21] Jesse Engel et al. "Adversarial Audio Synthesis". In: *International Conference on Learning Representations (ICLR)*. New Orleans, USA, 2019.

[22] Xin Wang, Shinji Takaki, and Junichi Yamagishi. "Autoregressive neural fo model for statistical parametric speech synthesis". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.8 (2018), pp. 1406–1419.

[23] Lingwei Meng et al. "Autoregressive Speech Synthesis without Vector Quantization". In: *arXiv preprint arXiv:2407.08551* (2024). {Last Accessed Date: 16th August, 2024}.

[24] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)". In: *University of Edinburgh. The Centre for Speech Technology Research (CSTR)* (2019), pp. 271–350.

[25] Heiga Zen et al. "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech". In: *INTERSPEECH*. Graz, Austria. 2019, pp. 1526–1530.

[26] Jörgen Valk and Tanel Alumäe. "VoxLingua107: A Dataset for Spoken Language Recognition". In: *IEEE Spoken Language Technology Workshop (SLT)*. Shenzhen, China, 2021, pp. 652–658.

[27] Andrew Estornell, Sanmay Das, and Yevgeniy Vorobeychik. "Deception through half-truths". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 34. 06. 2020, New York, USA, pp. 10110–10117.

[28] Nicolas Müller et al. "Does Audio Deepfake Detection Generalize?" In: *INTERSPEECH* (2022, Incheon, Korea), pp. 2783–2787.

[29] Junichi Yamagishi et al. "ASVSpoof 2021: Accelerating progress in spoofed and deepfake speech detection". In: *ASVSpoof 2021 Workshop-Automatic Speaker Verification and Spoofing Coutermeasures Challenge*. Singapore (virtual). 2021, pp. 47–54.

Appendix A

Experimental Setup

We employed two different types of features, and four different types of features for conducting experiments in this study. All the features selected were optimized in terms of dimensions. We also aim to analyze the effect of static vs. dynamic features.

Features Used

1. **MFCC:** Mel-Frequency Cepstral Coefficients (MFCCs) are widely used features in speech processing that capture the short-term power spectrum of a sound. MFCCs are derived by mapping the Fourier transform of a signal onto the mel scale, which approximates human auditory perception, emphasizing frequencies that humans are more sensitive to. The process typically involves dividing the speech signal into overlapping frames, applying a window function, performing a Fourier transform, mapping to the mel scale, taking the logarithm, and finally applying the Discrete Cosine Transform (DCT) to obtain the coefficients. We extracted MFCCs with a dimension of 20, the process captures the most essential features of the speech signal over each frame, frame length was taken as 25 ms with 50 % overlap, preserving the important phonetic details while reducing data redundancy.
2. **LFCC**: The spectral characteristics of audio signals are represented by linear frequency cepstral coefficients (LFCC). They are produced by applying a Fourier transform to the audio frames and visualizing the spectrum on a linear frequency scale. The energy distribution across multiple frequencies in a signal is captured by LFCC. They are often used for voice and audio processing tasks including speech recognition, music analysis, and most recently voice anti-spoofing. In contrast to MFCC, which uses a logarithmic Mel scale, a linear perspective on the frequency content is provided by LFCC.

Classifier Used

1. **CNN**: For this study, we employed CNN as pattern classifier because it captures spatial and temporal dependencies in the audio signals.The CNN architecture was built with a sigmoidal activation layer, and 3 ReLU activation layers. CNN consists of five convolution blocks and three fully-connected layers. Each layer is made up of 2-D convolution layers, a ReLU activation layer, and a batch normalization layer. At the end of each layer, max-pooling is

used to downsample feature maps. The final dense layer has a single unit with a sigmoid activation function, producing a binary classification output (0 or 1) that indicates whether the input belongs to class 0 or 1. Learning rate was taken as 0.003 and optimizer was chosen to be Adam. Input shape was taken to 20 x time_series x 1. Learning rate was selected as 0.003, with batch-size of 64. Adams optimizer were used for this paper.

2. **ResNet50**: ResNet contains four blocks within each block. The first block has three convolutional layers, followed by four, six, and three convolutional layers, respectively. Batch normalization and ReLU activation functions are applied after each convolutional layer. After the main blocks, there is a global average pooling layer that reduces the spatial dimen- sions of the feature maps. This is followed by a fully-connected layer with a softmax activation function, which produces the final output probabilities for different classes. This architecture is also known as ResNet-50.

3. **BiLSTM**: Bidirectional Long Short-Term Memory (Bi-LSTM) is a type of Recurrent Neural Network (RNN) that is commonly used in sequence modeling tasks, such as natural language processing and speech recognition. Bi-LSTM is an extension of the conventional LSTM architecture and performs both forward and backward processing of the input sequence, allowing it to gather information from both previous and future time steps. For this study, three Bi-LSTM layers were used, each consisting of 128 units, with a dropout of 10 % at the end of each layer. Finally, a dense layer with 155 units, and a softmax activation function was used as the output layer for classification. The BiLSTM is same as our previously employed one in Transfer Learning Using Whisper for Dysarthric Automatic Speech Recognition.

4. **BiGRU**: The BiGRU classifier is constructed to capture both forward and backward dependencies in sequential input data, which is beneficial for tasks such as audio classification. The model is initialized with an input size corresponding to the number of features per time step, and the hidden units define the size of the hidden states in the GRU layers. The network consists of multiple GRU layers (‘num_layers‘), with a bidirectional configuration that processes the input in both forward and backward directions. After the input is passed through the BiGRU layers, the forward and backward hidden states are concatenated to form a combined representation. Specifically, the last hidden state from the forward GRU and the first hidden state from the backward GRU are concatenated along the feature dimension to capture both temporal perspectives. This concatenated hidden state is then passed through a dropout layer, with a specified dropout rate (e.g., 0.255), to reduce overfitting. Finally, the combined features are fed into a fully connected layer that maps the hidden representation to the output classes, with the output dimension corresponding to the number of classes. This architecture enables the model to effectively leverage the temporal structure of the data for classification tasks.

Appendix B

Both the features were employed for cross training evaluation of the dataset, in particular BiLSTM classifier. Authors choose BiLSTM as a classifier, as it was able to obtain highest accuracy on track T2. As we can observe in Table 3, accuracy drops below 50 % and remains around 33 % due to models misclassifying deepfake audio as real audio. On the other hand, the model trained on proposed dataset (MLADDC) have accuracy almost above 50 % for each dataset, indicating the correct classification of the audio when the model trained on the proposed dataset. Current proposed system employs basic speech processing features such as, MFCC and LFCC, which if improved to advance features, such as, modified group delay (MGDF), or residual based (LPR), which may improve accuracy of model. Also limitations of current work include speech processing based methods for classification, which can be improved by employing other pre trained model based features such as, Whisper, wav2vec2.0, HuBERT and many more. Also classifiers can be empowered to advance classifiers and end to end models such as WaveNet, and AASIST.

It can be observed in Table 3, that the training on MLADDC dataset (T2) results in better testing accuracy, i.e., 67.92 % when tested on ITW dataset. On the other hand, when model trained on different existing datasets, i.e., FoR, ITW, and ASVSpoof, the testing accuracy is lower for unknown data testing (testing on MLADDC T2). This results may be due to multilingual data in proposed dataset, as well as fine generated deepfakes in the proposed dataset. On the other hand, when model is trained on proposed dataset and is tested on ASVSpoof dataset, results of LFCC features are upto 86.21 % which are much better as compared to other results. Such results on cross training are important for proving superiority of proposed dataset over existing datasets in various aspects.

Table 3: Cross training results on T2 Track of MLADDC dataset.

Train dataset	Test dataset	MFCC	LFCC
FoR	MLADDC (T2)	37.31	56.23
ITW	MLADDC (T2)	33.33	59.19
ASVSpoof	MLADDC (T2)	33.84	33.34
MLADDC (T2)	FoR	45.66	48.87
MLADDC (T2)	ITW	62.81	67.92
MLADDC (T2)	ASVSpoof	41.49	86.21

Table 4 displays the data statistics of selected data from VoxLingua107 dataset, and balancing of dataset.

Table 4: Original and Selected audios from VoxLingua107 Dataset.

Original/Selected	Language	Language Label	0-5	5-10	10-15	15-20	20+	Total Time (in Hours)
Original	Russian	ru	2028	8860	7044	5844	22	73
Selected	∼	∼	1000	1000	1000	1000	-	11.39
Original	French	fr	4234	9465	6248	4495	7	67
Selected	∼	∼	1000	1000	1000	1000	-	11.38
Original	Arabic	ar	3950	8422	5390	3914	6	59
Selected	∼	∼	1000	1000	1000	1000	-	11.34
Original	Spanish	es	988	5117	3817	2941	3	39
Selected	∼	∼	988	1004	1004	1004	-	11.37
Original	Vietnamese	vi	6861	12292	5169	3039	5	64
Selected	∼	∼	1000	1000	1000	1000	-	11.28
Original	Mandarin Chinese	zh	3243	6220	3861	3004	0	44
Selected	∼	∼	1000	1000	1000	1000	-	11.37
Original	English	en	1232	5953	4824	3874	2	49
Selected	∼	∼	1000	1000	1000	1000	-	11.32
Original	Hindi	hi	5492	11908	7382	5240	8	81
Selected	∼	∼	1000	1000	1000	1000	-	11.42
Original	Portuguese	pt	4572	9725	5764	4153	10	64
Selected	∼	∼	1000	1000	1000	1000	-	11.37
Original	Sanskrit	sa	2575	3978	938	328	0	15
Selected	∼	∼	1367	1367	938	328	-	8.92
Original	Bahasa Indonesia	id	3880	7399	3251	1980	0	40
Selected	∼	∼	1000	1000	1000	1000	-	11.35
Original	Bengali	bn	4433	8930	4861	3195	0	55
Selected	∼	∼	1000	1000	1000	1000	-	11.36
Original	Finnish	fi	913	4443	3249	2532	0	33
Selected	∼	∼	913	1029	1029	1029	-	11.52
Original	Japanese	ja	4948	9262	4879	3218	0	56
Selected	∼	∼	1000	1000	1000	1000	-	11.34
Original	Gujarati	gu	3766	7290	3998	2842	0	46
Selected	∼	∼	1000	1000	1000	1000	-	11.36
Original	Tamil	ta	3743	7679	4486	3267	0	51
Selected	∼	∼	1000	1000	1000	1000	-	11.4
Original	Punjabi	pa	4367	9098	4549	3078	0	54
Selected	∼	∼	1000	1000	1000	1000	-	11.36
Original	Urdu	ur	1254	4817	4011	3571	0	42
Selected	∼	∼	1000	1000	1000	1000	-	11.35
Original	Swedish	sv	2387	5136	3080	2174	0	34
Selected	∼	∼	1000	1000	1000	1000	-	11.39
Original	German	de	885	4981	3986	3012	0	39
Selected	∼	∼	885	1038	1038	1039	-	11.54