DFADD: THE DIFFUSION AND FLOW-MATCHING BASED AUDIO DEEPFAKE DATASET

Jiawei Du[1*], *I-Ming Lin*[1*], *I-Hsiang Chiu*[3*], *Xuanjun Chen*[2], *Haibin Wu*[2], *Wenze Ren*[2], *Yu Tsao*[4], *Hung-yi Lee*[2], *Jyh-Shing Roger Jang*[1]

[1]Department of Computer Science Information Engineering, National Taiwan University
[2]Graduate Institute of Communication Engineering, National Taiwan University
[3]Department of Electrical Engineering, National Taiwan University
[4]Academia Sinica, Taiwan

ABSTRACT

Mainstream zero-shot TTS production systems like Voicebox and Seed-TTS achieve human parity speech by leveraging Flow-matching and Diffusion models, respectively. Unfortunately, human-level audio synthesis leads to identity misuse and information security issues. Currently, many anti-spoofing models have been developed against deepfake audio. However, the efficacy of current state-of-the-art anti-spoofing models in countering audio synthesized by diffusion and flow-matching based TTS systems remains unknown. In this paper, we proposed the Diffusion and Flow-matching based Audio Deepfake (DFADD) dataset. The DFADD dataset collected the deepfake audio based on advanced diffusion and flow-matching TTS models. Additionally, we reveal that current anti-spoofing models lack sufficient robustness against highly human-like audio generated by diffusion and flow-matching TTS systems. The proposed DFADD dataset addresses this gap and provides a valuable resource for developing more resilient anti-spoofing models.

***Index Terms*—** dataset, deepfake detection, anti-spoofing, text-to-speech

1. INTRODUCTION

Text-to-speech (TTS) aims to generate natural and understandable audio based on given text content [1]. Tacotron 1/2 [2, 3] are early RNN-based TTS systems that significantly improved speech quality compared to previous methods. Transformer-based TTS systems [4–7] excel at modeling long-dependency speech and text sequences. Fastspeech [8,9] enhanced the robustness of TTS-generated audio by reducing word skipping and repetition with an external aligner. Glow-TTS [10] is a flow-based model that searches the most likely monotonic alignment between text and speech latent representations without needing external guidance. Despite their satisfactory performance, current diffusion and flow matching [11, 12] based models achieve better naturalness, speaker similarity, and sound quality. Diff-TTS [13] is one of the first diffusion-based TTS models, using a denoising diffusion framework to convert noisy signals into Mel-spectrograms to generate high-fidelity audio. In addition, diffusion-based models can produce audio quality that is indistinguishable from human speech, even replicating emotions and styles to a lifelike degree [14–22]. Flow matching (FM) based models primarily accelerate training and inference speed. They enable accurate synthesis with fewer steps [23–25]. However, the above-mentioned advancements in TTS technology also raise security concerns, as they provide malicious attackers with new speech synthesis tools that can lead to large-scale misuse.

Spoof detection [26–32] aims to distinguish genuine and spoofed utterances. To advance the development of anti-spoofing models, a large number of anti-spoofing challenges and datasets have been proposed so far [27, 33–39]. In recent years, significant progress has been made in developing high-performance anti-spoofing models for traditional speech synthesis systems. However, diffusion and FM based models are relatively new, and it remains uncertain whether the most advanced anti-spoofing models can effectively counter these types of synthetic speech.

In this paper, we introduce the Diffusion and Flow-matching based Audio Deepfake Dataset (DFADD), which comprehensively collects various advanced Diffusion and Flow-matching TTS models. The DFADD dataset comprises five diverse and mainstream open-source Diffusion and FM based TTS models. Additionally, we conduct a comprehensive analysis, meticulously evaluating the effectiveness of cutting-edge anti-spoofing models when confronted with synthesized speech generated by these advanced Diffusion and Flow-matching TTS models. Moreover, we utilize the DFADD dataset to develop significantly enhanced anti-spoofing models for effectively detecting spoofed audio generated by diffusion or flow matching based TTS systems.

We observe that: (1) Models trained on the ASVspoof dataset face challenges in detecting speech clips generated by advanced diffusion and FM based TTS systems. (2) Our

*equal first contribution

proposed DFADD dataset significantly improves the models' ability to handle synthesized speech from current various state-of-the-art (SOTA) diffusion and FM based TTS systems (Compared to training on ASVspoof datasets, the models trained on DFADD subsets achieve an average equal error rate (EER) reduction of over 47%).

We will soon release the data [1] and hope this study and the DFADD dataset can reduce malicious attacks from advanced diffusion and FM based TTS systems. Our audio samples can be found on the demo page [2].

2. RELATED WORK

The development of anti-spoofing models requires extensive and robust datasets as training data. We will elaborate on existing training datasets and defense models in related work.

2.1. Audio Anti-Spoofing Dataset

Several audio anti-spoofing datasets have been released using various deepfake techniques, including generative models, partial spoofs, multimodal deepfakes, and multi-language spoofing audio. We introduce the audio deepfake datasets containing English speakers, with details shown in Table 1.

ASVspoof19-LA [34] contains spoofed audios generated from TTS and Voice Conversion (VC). All of them are from the VCTK dataset [40]. The ASVspoof2019-LA evaluation set contains 13 unknown TTS and VC algorithm-generated spoofed speech to verify the generalization of anti-spoofing detection algorithms.

ASVspoof21-DF [36] is an audio dataset generated by more than 100 TTS and VC methods and includes different compression algorithms and source domains. ASVspoof-DF simulates the processing of different lossy codecs in real situations when handling the dataset.

ASVspoof21-LA [36] includes the training and development sets of ASVspoof2019-LA and evaluation set of ASVspoof2021-LA. The evaluation set of ASVspoof2021-LA has been processed by real phone systems with various codecs, transmission channels, bit rates, and sampling rates.

WaveFake [37] has 117,985 spoofed speech clips. The bonafide speech clips are collected from LJspeech[3] and JSUT [41] dataset. Its 10 subsets are generated from 5 different GAN-based TTS models and one flow-based generative model across two languages.

In-The-Wild (ITW) [38] contains 37.9 hours of audio recordings of celebrities and politicians, 17.2 hours of which are faked. There may be background noise since the recorded audio is publicly available on the Internet.

TIMIT-TTS [42] is an audio dataset that uses the VidTIMIT [43] dataset as a reference, which can be used for multimodal synthetic media detection or as an audio deepfake dataset only. The video in VidTIMIT is split into audio content and visual content, and the deepfake audio is generated through three steps applied to the audio content. First, the original audio is transcribed into text by a speech-to-text algorithm. Second, spoof audio is generated from text using 12 existing TTS models. Finally, the spoof audio is synchronized with the original audio.

MLADD [39] is a multi-language audio anti-spoofing dataset, utilizing 54 TTS models built from 21 different architectures, and generating 163.9 hours of synthetic voice across 23 different languages. The dataset is introduced because of the language bias present in deepfake audio datasets, most of them predominantly consist of English speech. By incorporating multilingual audio samples, detection models can enhance their ability to generalize across datasets, thus more advantageously combat audio spoofing and deepfakes.

However, the aforementioned datasets do not consider the newly emerged diffusion and FM based TTS models.

2.2. Anti-spoofing

The anti-spoofing model is designed to differentiate between genuine and spoofed utterances, mitigating the impact of synthetic speech. AASIST [44] is one of the SOTA anti-spoofing models. It takes Rawnet2 [45] as its speech encoder to extract features and employs two graph modules for spectral and temporal domain modeling. In addition, it utilizes max graph operations with heterogeneous graph modeling via HS-GAL [46] layers, and achieves final classification through an output layer after element-wise maximum and node value aggregation. AASIST-L uses several model compression techniques to reduce its size by 70% compared to the AASIST model while keeping the overall architecture unchanged, alleviating the overfitting issues. We use AASIST-L as our backbone trained on different subsets of DFADD. This is because AASIST-L is less prone to overfitting on ASVspoof2019 dataset. More detailed experimental setups will be explained in **Section 4.1**.

3. DFADD DATASET

This section describes the creation of the DFADD dataset, its design principles, and the reasoning behind its development. The pipeline for generating our dataset is shown in Fig.1. It consists of two stages: input selection and text-to-speech synthesis. In the input selection stage, we obtain the target speaker prompts s and text prompts t. In the text-to-speech synthesis stage, each selected speaker prompt s_i and selected text prompt t_i will be input to the TTS model to generate spoofed audio. We used 5 different diffusion and FM-based TTS models, which will be introduced in Section.3.2 There is a one-to-one correspondence between bonafide and spoofed speakers. In other words, both bonafide and spoofed speakers

[1] https://github.com/isjwdu/DFADD

[2] https://github.com/DFADD-Dataset/DFADD_demo_pages/

[3] https://keithito.com/LJ-Speech-Dataset/

Table 1: Comparison of DFADD with other deepfake datasets containing English speakers [47]. None means no detailed information is provided.

Dataset	Type	Methods	Speakers	Language	Bonafide	Spoofed
ASVspoof19 LA	TTS VC	19	48	English	10,256	90,192
ASVspoof21 DF	TTS VC	100+	93	English	14,869	519,059
ASVspoof21 LA	TTS VC	19	67	English	14,816	133,360
WaveFake	TTS	7	2	English Japanese	0	117,985
TIMIT	TTS	12	46	English	0	5,160
ITW	None	None	58	English	19,963	11,816
MLAAD	TTS	54	None	English, other 22	0	76,000
DFADD	TTS	5	109	English	44,455	163,500

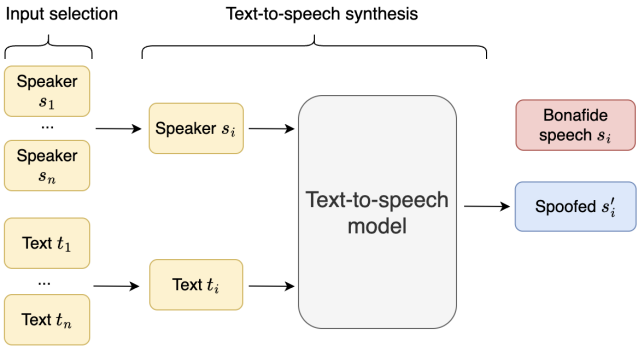

Fig. 1: The pipeline of data generation for DFADD.

have the same speaker identity, with the difference being that one is authentic while the other is synthesized through TTS.

3.1. Input selection

Text selection. To prevent text data leakage from models pre-trained on VCTK, we avoided using the same text prompts from VCTK, and instead used LJspeech to get our text prompts. To ensure the audio duration and quality are similar to VCTK samples, we removed sentences with complex words (such as names and special nouns) and selected sentences with 5 to 10 words. Ultimately, we chose 300 sentences that met these criteria to serve as text prompt inputs for the subsequent stage of speech synthesis in TTS systems.

Speaker selection. We use the VCTK dataset, which includes 109 speakers, as our input. Each speaker undergoes inference on the 300 sentences selected during **Text selection** to generate the spoofed audio in the synthesis stage.

3.2. Text-to-speech model

We leveraged the released checkpoints trained on VCTK or zero-shot models using diffusion and FM based methods. We selected 5 different TTS systems as the backbone for our dataset generation. Diffusion-based TTS systems include Grad-TTS [48], NaturalSpeech 2 [14], and Style-TTS 2 [15]. FM-based TTS systems include Matcha-TTS [24] and PFlow-TTS [25]. To simplify the description, let "D∼" stands for Diffusion, and "F∼" represents Flow-matching.

3.2.1. Diffusion-based Text-to-speech

Diffusion-based TTS models introduce noise into audio features and progressively denoise them to produce high-quality speech features or waveforms. Their superior performance is due to their ability to model complex data distributions with fine-grained control, reducing artifacts, controlling speech emotions, and stylizing speech from text, resulting in highly natural audio.

D1. Grad-TTS [48]. It encodes the text into features and aligns with the text input using the Monotonic Alignment Search algorithm, creating a monotonic mapping between the text and the mel-spectrogram. The diffusion process generates mel-spectrograms from Gaussian noise, guided by a noise scheduling function and reversed through time-based inference to reconstruct the target distribution gradually.

D2. NaturalSpeech 2 [14]. In the training process, NaturalSpeech 2 converts the input speech waveform into quantized latent vectors. Then, a diffusion model predicts these latent vectors from the text input. The model conditions on the output of phoneme encoder, duration predictor, and pitch predictor. During inference, the diffusion model first generates the latent vectors from the text or phoneme sequence and then converts these latent vectors into the final speech waveform using the decoder in the neural audio codec.

D3. Style-TTS 2 [15]. It uses a text encoder to convert the input text into phoneme representations. The diffusion model samples a style vector from a latent random variable conditioned on the input text, modeling diverse speech styles. The style vector is fed into the speech decoder, which combines it with the phoneme representations, pitch curve, and energy curve to generate the final speech waveform directly. To achieve efficient generation, StyleTTS 2 uses large pre-trained speech language models (e.g., WavLM) as discriminators and introduces differentiable duration modeling to enhance speech naturalness and generation quality.

3.2.2. Flow-matching based Text-to-speech

FM-based TTS models further enhance the efficiency. It eliminates the need for numerically solving the reverse-time stochastic differential equation, which requires many steps. After obtaining the acoustic features, these models aim to directly model the vector field implied by an arbitrary ordinary differential equation (ODE). All FM-based models consider linearized sampling trajectories and minimize transmission costs from data distribution or noise, thereby finding

a more straightforward path from source to target, resulting in higher-quality synthesis with fewer steps.

F1. Matcha-TTS [24]. Matcha-TTS employs a text encoder to convert the input text into a sequence of phonemes, capturing textual information. A duration predictor estimates the duration for each phoneme to ensure the synthesized speech is synchronized using the input text. It then employs a conditional flow matching approach to train the whole model, optimizing the path from latent space to the data distribution, thereby reducing the number of steps needed for synthesis. Finally, the duration predictor's output with the diffusion process generates mel-spectrograms from noise and uses a neural decoder to convert these mel-spectrograms into the final speech waveforms.

F2. PFlow-TTS [25]. PFlow-TTS is a zero-shot TTS model that generates high-quality speech for unseen speakers using minimal training data. It consists of a speech-prompted text encoder that combines a short speech prompt with text input to produce a speaker-conditioned text representation. This representation is used by the flow-matching generative decoder to synthesize speech, converting the text to a mel-spectrogram and then to a waveform. PFlow-TTS achieves superior speed and data efficiency by avoiding autoregressive components and neural codecs, using flow matching for faster and more direct speech synthesis. This method provides significant improvements in inference speed and speaker adaptation, maintaining high speech quality with reduced data and simpler training.

3.2.3. Text-to-speech synthesis

For D2, D3, and F1, we used models pre-trained on VCTK to perform inference on different speakers. We trained D1 and F2 from scratch to adapt to VCTK speakers. F2 is an unofficial implementation. F1 is the only officially open-sourced TTS system that uses the FM method and supports inference for VCTK speakers.

D1. Grad-TTS audio synthesis. During the training phase, we followed the default Grad-TTS hyperparameter settings. We trained Grad-TTS for 1000 epochs on a V100-32G GPU, with a batch size of 16 and a sample rate of 22,050 Hz. During the inference phase, we replaced the vocoder provided by Grad-TTS with HiFi-GAN, which is pre-trained on VCTK. Additionally, we changed the diffusion time steps and temperature to 70 and 3, respectively.

D2. NaturalSpeech 2 audio synthesis. We used a zero-shot approach with a prompt speech and the text mentioned in **Input selection** to generate a spoofed speech. The prompt speech we used is utterance number 016 of bonafide audio from each speaker. We use the unofficial checkpoint [4] pre-trained on the VCTK for 306K steps with V100-32G GPU.

D3. Style-TTS 2 audio synthesis. StyleTTS2 also uses a zero-shot approach with a prompt speech and text to generate spoofed D3 subset, similar to NaturalSpeech 2. We use the checkpoint pre-trained on the LibriTTS dataset and set the parameters α and β in Style-TTS 2 both to 0, making the generated spoof speech as similar as possible to the original.

F1. Matcha-TTS audio synthesis. We used the official checkpoint pre-trained on VCTK to generate the F1 subset. For inference, we used a V100 GPU, with the temperature set to 0.667 and the ODE step set to 10.

F2. PFlow-TTS audio synthesis. During the training phase, we follow PFlow-TTS's default hyperparameter settings. We trained for 1100 epochs on a GPU V100 32G, and the batch size was 16. During the inference phase, we replace the vocoder provided by the unofficial PFlow-TTS[5] with HiFi-GAN [49], which is pre-trained on VCTK. Since some poor-quality real audio is removed from the VCTK, we use the bonafide audio with utterance number 013 as the prompt speech of speakers p292 and p318. For other speakers, the utterance number of prompt speech is 003. Our ODE steps and temperature are the same as F1's settings.

3.3. Dataset comparison

We divided 109 speakers into three speaker-disjoint sets for training, validation, and testing. Speakers for validation data are p226 and p229, while speakers p227 and p228 are used for testing. The remaining speakers are allocated for training. The sample rate of all audio files is set to 16,000 Hz.

We generate 163,500 TTS-based spoofed speech clips totaling 179.88 hours from the bonafide speech clips, with an average length of 4.01 seconds. Table 2 shows a detailed summary of subsets. In comparison with existing mainstream audio anti-spoofing datasets, which primarily use TTS and VC methods, our dataset focuses solely on TTS systems. While previous TTS anti-spoofing datasets were generated using traditional neural network methods (e.g., Flow-based, GAN-based), these methods are inferior to diffusion-based and FM-based approaches in generation quality. Furthermore, DFADD features the largest number of speakers among anti-spoofing datasets in Table 1, and the speech clips we generated far exceed the number of spoofed speech clips in other TTS-only datasets.

4. EXPERIMENTAL SETUP

4.1. Anti-spoofing model setup

We use one of SOTA deepfake detection models, AASIST-L [44] [6], as our backbone for training anti-spoofing models. We chose AASIST-L because it is less prone to overfitting on the ASVspoof dataset.

For the ASVspoof dataset, we utilize the author's released checkpoints after their thorough hyperparameter search. For

[4]https://github.com/CODEJIN/NaturalSpeech2

[5]https://github.com/p0p4k/pflowtts_pytorch

[6]https://github.com/clovaai/aasist

Table 2: Comparison between DFADD different generation pipelines. D1 refers to the GradTTS. D2 signifies NaturalSpeech 2. D3 represents the StyleTTS 2. F1 means the Matcha-TTS. F2 represents the PFlow-TTS. Train, valid, and test represent the average duration (seconds), respectively.

Subsets	Methods	Source	train	valid	test
D1	Diffusion	Train	3.06	3.06	3.06
D2	Diffusion	Pretrain	5.63	5.67	5.84
D3	Diffusion	Pretrain	3.84	3.97	4.10
F1	Flow matching	Pretrain	2.98	2.88	3.10
F2	Flow matching	Train	4.27	4.24	4.42
DFADD	-	-	3.83	3.85	4.01

DFADD, our model used the Adam optimizer with a learning rate of 0.001 and a batch size of 24, trained on a V100 32G GPU. During training, one of the DFADD subsets is used as spoofed audio, and the corresponding bonafide VCTK utterances are combined as training data. We use the corresponding DFADD validation subset for model selection.

4.2. Evaluation setup

We consider two evaluation scenarios: the seen scenario and the unseen scenario. (1) The seen scenario involves evaluating the anti-spoofing model on the evaluation set of each DFADD subset. This means the model has been exposed to the same distribution of datasets during training and has learned the features of the corresponding subset. (2) The unseen scenario involves evaluating the models on audio samples collected from demo pages of various TTS systems. Since the models were trained on the DFADD subsets, they did not learn the features from these collected audio samples. These TTS systems in unseen scenario include VoiceBox [23], VoiceFlow [50] NaturalSpeech 3 [51], CMTTS [52], DiffProsody [53], and DiffAR [54]. VoiceFlow and VoiceBox use FM-based methods, while the others use diffusion-based methods. The ground truth data comes from actual recordings or speaker prompts, and the model-generated audio is classified as spoofed data. By evaluating on these unseen datasets, we assess the generalization performance of our models.

5. EXPERIMENTS RESULTS

5.1. Audio quality assessment

We leverage the UT-MOS [55] to assess the quality of our synthesized audio in the DFADD dataset. MOS usually ranges from 1 to 5, where higher scores represent better natural synthesis quality. The detailed MOS distribution of DFADD is shown in Fig-2. Over 97% of DFADD speech clips (including bonafide and spoofed) have an MOS of 3.0 or above. The quality of the spoofed audio generated by D3 is especially natural, with most MOS greater than 4.0, making

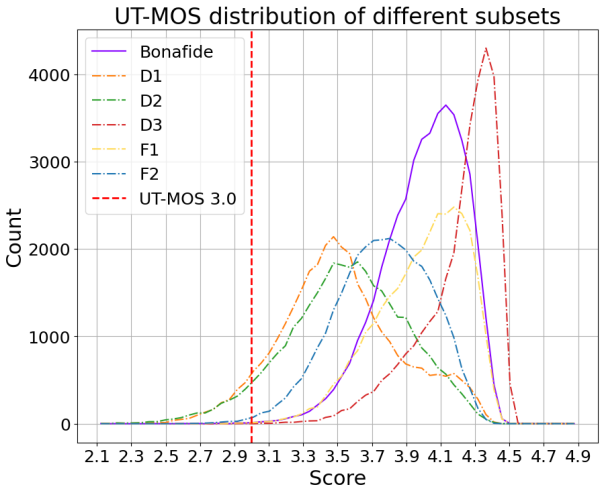

Fig. 2: MOS distribution of spoofed audio generated by different TTS models (higher means more natural).

it comparable to bonafide audio. While UT-MOS has known limitations and inherent biases [56–58], the results still indicate that most quality of spoofed audio samples in DFADD are close to genuine audio, highlighting the potential misuse of diffusion and FM-based TTS models in malicious attacks.

5.2. Seen scenario cross-testing evaluation

The Fig. 3 shows the cross-testing results followed by the dataset splitting method in Section 3.3. The rows in the figure represent subsets of training data from DFADD and ASVspoof, while the columns correspond to subsets of testing data from DFADD. Our observations are as follows:

(1) Most testing subsets show significantly high EERs, typically above 30%, for the AASIST and AASIST-L models trained on the ASVspoof dataset. This indicates that these models struggle to distinguish speech generated by diffusion-based and FM-based TTS systems.

(2) From the perspective of the horizontal axis, models training on subsets from the FM-based TTS pipeline (F1, F2) perform very well on the diffusion-based test subsets except for D3, with EERs very close to 0. This indicates that the anti-spoofing model trained on the FM-based audio deepfake dataset has better generalization performance compared to the model trained on the diffusion-based audio deepfake dataset.

(3) From the perspective of the vertical axis, training on a particular DFADD subset consistently results in the lowest EER for its corresponding testing subset. For instance, training on subset D1 yields the lowest EER when testing on D1, and similarly for subset F1. Additionally, D3 is the most difficult subset to fit in each training and testing scenario. This may be because the audio of D3 is so realistic that the models trained on other subsets cannot distinguish it. This is also indirectly indicated by the higher MOS scores of D3 compared to the other subsets in Fig.2.

Table 3: Performance comparison of spoofed speech detection (EER) between models trained on ASVspoof and DFADD. Models surpassing those trained on ASVspoof are emphasized in bold. The top-performing models feature a gray background.

	ASVspoof (All)		DFADD				
	AASIST	AASIST-L	D1	D2	D3	F1	F2
VoiceBox [23]	42.59	47.62	53.77	64.70	50.13	**36.69**	60.80
VoiceFlow [50]	50.41	41.33	**34.70**	**33.06**	**33.06**	42.96	**24.80**
NaturalSpeech3 [51]	24.50	25.50	31.63	59.69	62.25	**18.38**	**24.50**
CMTTS [52]	56.54	43.46	**20.26**	**10.13**	**10.13**	**0.00**	**0.00**
DiffProsody [53]	37.50	35.94	62.50	**28.13**	**25.00**	**25.00**	**25.00**
DiffAR [54]	53.72	69.42	74.73	**50.53**	**27.39**	**25.27**	**4.26**
Average	44.21	43.88	46.26 (+2.38)	41.04 (-2.84)	34.66 (-9.22)	24.72(-19.16)	23.22 (-20.66)

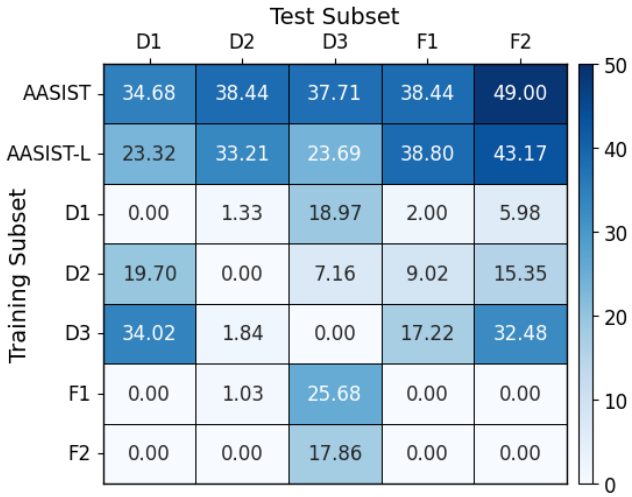

Fig. 3: Cross-testing EER results of anti-spoofing models on DFADD test subsets. The evaluation metric is equal error rate (EER), where lower is better.

5.3. Unseen scenario cross-testing evaluation

Table 3 presents the performance of AASIST-L models trained on various subsets and evaluated on the unseen scenario. The columns show the training sets used to develop each model. For the ASVspoof dataset, all available data were used to train two model variants: AASIST and AASIST-L. For the DFADD, each column indicates the training subset used for the AASIST-L model. The rows indicate the sources of the testing sets. The following observations were made:

(1) The anti-spoofing models trained with the ASVspoof dataset exhibit notably poor performance on the unseen evaluation dataset. This pronounced discrepancy likely arises because the ASVspoof dataset mainly contains speech clips generated by traditional TTS and VC methods, which differ significantly from the diffusion and FM based methods in DFADD. This difference highlights the urgent need for datasets generated by these advanced methods to improve the robustness of anti-spoofing models.

(2) From the perspective of each unseen evaluation set, the EER of the model trained on a single subset is generally lower than when trained on ASVSpoof in most unseen evaluation datasets. Notably, the CMTTS models show a significant decrease in EER regardless of the subset used for training. Additionally, models trained on FM-based TTS subsets exhibit the highest degree of generalizability, significantly reducing their EER in most unseen scenarios.

(3) From the perspective of average EERs on individual DFADD subsets, anti-spoofing models trained on DFADD subsets show a high effectiveness in detecting spoofing in unseen and similar methods (diffusion and FM based TTS) datasets. Specifically, the average EERs of models trained on F1 and F2 are reduced by 19.16 and 20.66, respectively, compared to the baseline AASIST-L (trained on ASVspoof). In addition, the reduction achieved by anti-spoofing models trained on FM-based audio samples is significantly greater than that achieved by models trained on diffusion-based subsets. These findings indicate that models trained on FM-based subsets exhibit better generalization capabilities.

6. CONCLUSION

In this study, we assembled DFADD, the first dataset that includes spoofed speech generated specifically using advanced diffusion and FM based TTS models. We verified that the spoofed audio generated by these models has a highly natural quality. Our extensive experiments demonstrate that anti-spoofing models trained on the ASVspoof dataset struggle to detect spoofs from diffusion and FM based TTS models, but the DFADD dataset significantly enhances their performance. The average EER of an anti-spoofing model on unseen scenarios was reduced by more than 47% due to train on DFADD subsets. All codes and data will soon be released to help resist malicious attacks from advanced diffusion and FM based speech synthesis systems.

7. ACKNOWLEDGEMENTS

This work was partially supported by the National Science and Technology Council, Taiwan (Grant no. NSTC 112-2634-F-002-005, Advanced Technologies for Designing Trustable AI Services). We also thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

8. REFERENCES

[1] Chenshuang Zhang et al., "A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai," *arXiv preprint arXiv:2303.13336*, vol. 2, pp. 2, 2023.

[2] Yuxuan Wang et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[3] Jonathan Shen et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[4] Ashish Vaswani et al., "Attention is all you need," *Proc. NeurIPS*, vol. 30, 2017.

[5] Naihan Li et al., "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019, vol. 33.

[6] Adrian Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *Proc. ICASSP*, 2021.

[7] Dan Lim et al., "Jdi-t: Jointly trained duration informed transformer for text-to-speech without explicit alignment," *arXiv preprint arXiv:2005.07799*, 2020.

[8] Yi Ren et al., "Fastspeech: Fast, robust and controllable text to speech," *Proc. NeurIPS*, vol. 32, 2019.

[9] Yi Ren et al., "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[10] Jaehyeon Kim et al., "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *Proc. NeurIPS*, vol. 33, 2020.

[11] Quan Dao et al., "Flow matching in latent space," *arXiv preprint arXiv:2307.08698*, 2023.

[12] Yaron Lipman et al., "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[13] Myeonghun Jeong et al., "Diff-tts: A denoising diffusion model for text-to-speech," *arXiv preprint arXiv:2104.01409*, 2021.

[14] Kai Shen et al., "Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers," *arXiv preprint arXiv:2304.09116*, 2023.

[15] Yinghao Aaron Li et al., "Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models," *Proc. NeurIPS*, vol. 36, 2024.

[16] Dong Zhang et al., "Speechgpt-gen: Scaling chain-of-information speech generation," *arXiv preprint arXiv:2401.13527*, 2024.

[17] Rongjie Huang et al., "Prodiff: Progressive fast diffusion model for high-quality text-to-speech," in *Proc. ACM Multimedia*, 2022.

[18] Heeseung Kim et al., "Guided-tts: A diffusion model for text-to-speech via classifier guidance," in *Proc. ICML*, 2022.

[19] Nanxin Chen et al., "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[20] Zhifeng Kong et al., "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[21] Max WY Lam et al., "Bddm: Bilateral denoising diffusion models for fast and high-quality speech synthesis," *arXiv preprint arXiv:2203.13508*, 2022.

[22] Sang-gil Lee et al., "Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior," *arXiv preprint arXiv:2106.06406*, 2021.

[23] Matthew Le et al., "Voicebox: Text-guided multilingual universal speech generation at scale," *Proc. NeurIPS*, vol. 36, 2024.

[24] Shivam Mehta et al., "Matcha-tts: A fast tts architecture with conditional flow matching," in *ICASSP*, 2024.

[25] Sungwon Kim et al., "P-flow: A fast and data-efficient zero-shot TTS through speech prompting," in *NeurIPS*, 2023.

[26] Haibin Wu et al., "The defender's perspective on automatic speaker verification: An overview," *arXiv preprint arXiv:2305.12804*, 2023.

[27] Zhizheng Wu et al., "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech*, 2015.

[28] Yen-Lun Liao et al., "Adversarial speaker distillation for countermeasure model on automatic speaker verification," *arXiv preprint arXiv:2203.17031*, 2022.

[29] Xuanjun Chen et al., "Singing voice graph modeling for singfake detection," *arXiv preprint arXiv:2406.03111*, 2024.

[30] Xuanjun Chen et al., "Neural codec-based adversarial sample detection for speaker verification," in *Interspeech 2024*, 2024.

[31] Z. Pan et al., "Attentive merging of hidden embeddings from pre-trained speech model for anti-spoofing detection," in *Interspeech 2024*, 2024.

[32] J. Li et al., "An initial investigation of neural replay simulator for over-the-air adversarial perturbations to automatic speaker verification," in *Proc. ICASSP*, 2024.

[33] Héctor Delgado et al., "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements ," in *Proc. Odyssey*, 2018.

[34] Andreas Nautsch et al., "Asvspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Trans. Biom. Behav. Identity Sci.*, 2021.

[35] Haibin Wu, Yuan Tseng, and Hung-yi Lee, "Codecfake: Enhancing anti-spoofing models against deepfake audios from codec-based speech synthesis systems," *arXiv preprint arXiv:2406.07237*, 2024.

[36] Xuechen Liu et al., "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *Proc. TASLP*, vol. 31, 2023.

[37] Joel Frank et al., "Wavefake: A data set to facilitate audio deepfake detection," *arXiv preprint arXiv:2111.02813*, 2021.

[38] Nicolas M Müller et al., "Does audio deepfake detection generalize?," *arXiv preprint arXiv:2203.16263*, 2022.

[39] Nicolas M Müller et al., "Mlaad: The multi-language audio anti-spoofing dataset," *arXiv preprint arXiv:2401.09512*, 2024.

[40] Junichi Yamagishi et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," in *Proc. CSTR*, 2019.

[41] Ryosuke Sonobe et al., "Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.

[42] Davide Salvi et al., "Timit-tts: A text-to-speech dataset for multimodal synthetic media detection," *IEEE Access*, 2023.

[43] Astik Biswas et al., "Vidtimit audio visual phoneme recognition using aam visual features and human auditory motivated acoustic wavelet features," in *Proc. ReTIS*, 2015.

[44] Jee-weon Jung et al., "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. ICASSP*, 2022.

[45] Hemlata Tak et al., "End-to-end anti-spoofing with rawnet2," in *Proc. ICASSP*, 2021.

[46] Xiao Wang et al., "Heterogeneous graph attention network," in *Proc. WWW*, 2019.

[47] Menglu Li et al., "Audio anti-spoofing detection: A survey," *arXiv preprint arXiv:2404.13914*, 2024.

[48] Vadim Popov et al., "Grad-tts: A diffusion probabilistic model for text-to-speech," in *Proc. ICML*, 2021.

[49] Jungil Kong et al., "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, 2020.

[50] Yiwei Guo et al., "Voiceflow: Efficient text-to-speech with rectified flow matching," in *Proc. ICASSP*, 2024.

[51] Zeqian Ju et al., "Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models," *arXiv preprint arXiv:2403.03100*, 2024.

[52] Xiang Li et al., "CM-TTS: Enhancing real time text-to-speech synthesis efficiency through weighted samplers and consistency models," in *Findings of NAACL*, 2024.

[53] Hyung-Seok Oh et al., "Diffprosody: Diffusion-based latent prosody generation for expressive speech synthesis with prosody conditional adversarial training," *Proc. TASLP*, vol. 32, 2024.

[54] Roi Benita et al., "Diffar: Denoising diffusion autoregressive model for raw speech waveform generation," *arXiv preprint arXiv:2310.01381*, 2023.

[55] Takaaki Saeki et al., "Utmos: Utokyo-sarulab system for voicemos challenge 2022," *arXiv preprint arXiv:2204.02152*, 2022.

[56] Slawomir Zielinski, Francis Rumsey, and Søren Bech, "On some biases encountered in modern audio quality listening tests-a review," *J. Audio Eng. Soc.*, 2008.

[57] Andrew Rosenberg et al., "Bias and statistical significance in evaluating speech synthesis with mean opinion scores.," in *Interspeech*, 2017.

[58] Erica Cooper et al., "Investigating range-equalizing bias in mean opinion score ratings of synthesized speech," *arXiv preprint arXiv:2305.10608*, 2023.