APSIPA Transactions on Signal and Information Processing, 2025, 14, e203

Original Paper

InaSAS: Benchmarking Indonesian Speech Antispoofing Systems

Candy Olivia Mawalim[1*], Sarah Azka Arief[2] and Dessi Puji Lestari[2]

[1] *Japan Advanced Institute of Science and Technology, Ishikawa, Japan*
[2] *Bandung Institute of Technology, Bandung, Indonesia*

ABSTRACT

Voice-based biometric systems are vulnerable to spoofing attacks, where attackers can deceive the systems with synthetic or replayed voice samples. To address this vulnerability, we introduce the InaSpoof-v1 dataset, which is a comprehensive benchmark for Indonesian language spoofing detection. We evaluate the state-of-the-art countermeasure models on this dataset, highlighting the challenges posed by the diversity of the Indonesian language and the impacts of demographic factors. Our experimental results demonstrate the effectiveness of the end-to-end AASIST model for synthesized speech attack countermeasures and residual networks (ResNet) for replay attack detection. To improve future systems, we emphasize the importance of considering demographic factors and addressing the challenges posed by real-world scenarios.

Keywords: Spoof countermeasure, Indonesian language, speech synthesis, replay attack

1 Introduction

Voice-based biometric systems have gained popularity in recent years because of their convenience and ease of use. However, these systems are vulnerable

*Corresponding author: candylim@jaist.ac.jp

Received 31 October 2024; revised 07 April 2025; accepted 21 April 2025
ISSN 2048-7703; DOI 10.1561/116.20240080

to spoofing attacks, where malicious actors attempt to deceive the systems by presenting artificial or synthetic voice samples [13]. Robust spoofing detection techniques or countermeasures are essential for mitigating these threats [14].

Voice-based spoofing attacks can be categorized into two main types: direct and indirect attacks. Direct attacks, which are also known as physical access attacks, involve presenting spoofed audio directly to the microphone of the target system. Indirect attacks, or logical access attacks, manipulate the internal processes of the system, such as its feature extraction or decision-making process, to bypass security measures. Spoofed speech samples can be generated through various techniques, including speech synthesis, voice conversion (VC), and recorded speech playback [42].

The development of reliable detection techniques has become a critical research area for addressing the growing concern related to audio deepfakes and spoofs. The ASVspoof challenge series, which spanned from 2015–2024 [45, 12, 42, 30, 11], has played a pivotal role in driving spoof detection advancements by progressively addressing various challenges. These challenges have attracted significant attention from researchers worldwide, leading to the development of numerous countermeasure techniques. However, a major remaining challenge is the generalization gap between controlled and real-world scenarios. Models that perform well in controlled environments often struggle to generalize to real-world conditions [34]. Additionally, the language bias exhibited by deepfake audio data, predominantly English data, limits the applicability of detection techniques to cases involving non-English languages.

Indonesia, which is a nation with over 700 living languages, boasts the second-highest linguistic diversity globally, accounting for almost 10% of the world's languages [27]. This linguistic richness, particularly the diverse dialects and unique phonetic characteristics of Indonesian [1], might present specific challenges for spoofing detection. While previous research has focused primarily on English-language spoofing detection, robust countermeasures are needed for other languages, including Indonesians. The development of robust Indonesian-language antispoofing systems can contribute to the overall improvement of automatic speaker verification (ASV) systems and their resilience against spoofing attacks.

This paper contributes to the advancement of Indonesian-language spoofing detection in the following ways.

- **Curating a reliable dataset:** We introduce the InaSpoof-v1 dataset, which is a comprehensive benchmark for evaluating Indonesian-language spoofing detection systems.
- **Developing and evaluating Indonesian antispoofing models:** We develop and evaluate a variety of antispoofing models, including traditional machine learning techniques and deep learning models.

- **Analyzing demographic factors:** We investigate the impacts of different demographic factors, such as gender and dialect, on the performance of spoofing detection systems.
- **Simulating real-world attacks:** We simulate real-world spoofing attacks using physical devices to evaluate the performance of countermeasure models under realistic conditions.

By addressing these objectives, this research contributes to the advancement of Indonesian-language spoofing detection. It provides valuable insights, serves as a resource for developing low-resource language datasets, and offers guidance for creating effective countermeasures with limited computational resources.

2 Related Works

Spoofing attacks can occur at different stages of the ASV process, including at the microphone and transmission levels [43]. These attacks may involve techniques such as voice replay, speech synthesis, and VC. To address this issue, researchers have explored various countermeasures, including advanced feature extraction and machine learning algorithms [20].

The initial efforts to address spoofing attacks in ASV scenarios began with the Interspeech 2013 special session on spoofing and antispoofing (SAS), which focused on speech synthesis and VC attacks [44]. Building on this initial work, the ASVspoof challenge series emerged as a significant driver for research and development in this area. These challenges, which spanned from 2015–2024, have provided standardized datasets and evaluation metrics, enabling researchers to develop and compare various countermeasure techniques.

Early ASVspoof challenges focused primarily on synthetic speech and VC attacks, often under controlled conditions [45, 12]. However, as deep learning-based speech synthesis techniques such as generative adversarial networks (GANs) and WaveNet have advanced, more sophisticated spoofing attacks have emerged. The ASVspoof 2019 challenge introduced a new metric, the tandem decision cost function (t-DCF), which provides a more comprehensive evaluation metric that considers both countermeasure detection accuracy and system decision costs [42]. The ASVspoof 2021 challenge further pushed boundaries by incorporating deepfake speech detection [30]. While significant progress has been made in logical and deepfake attack detection scenarios, physical access attacks remain challenging to detect due to the variability of real-world environments [47].

The latest ASVspoof 5 challenge further expanded the scope of spoofing detection by incorporating a larger and more diverse dataset that included crowdsourced data and adversarial attacks [11]. It also introduced new met-

rics for evaluating both standalone countermeasures and integrated ASVs. Despite the challenge complexities, a significant number of participants successfully developed systems that surpassed the performance of baseline models. For example, contemporary approaches frequently leverage state-of-the-art self-supervised learning (SSL) models like wav2vec 2.0 [6], WavLM [9], and HuBERT [17], which have become dominant in feature extraction. These SSL models outperform traditional hand-crafted features, such as Mel-frequency cepstral coefficients (MFCC), by capturing richer latent representations [41], [37], [46]. However, the challenge also highlighted the importance of performing score calibration for practical deployment purposes [41].

In addition to the ASVspoof challenge series, the Audio Deepfake Detection Challenge (ADD) series has emerged as a significant platform for advancing research in the audio deepfake detection field. ADD 2022 focused on tasks such as low-quality and partially fake audio detection. While this challenge led to progress, it had limitations, including a focus on binary classification [48]. To address these limitations, ADD 2023 introduced more complex tasks, such as manipulated segment localization and source identification, encouraging the development of more sophisticated detection techniques [49].

Several voice-based antispoofing datasets have been released to train and evaluate models, primarily focusing on the English and Chinese languages (e.g., the ASVspoof and ADD challenges). While these datasets have significantly advanced the field, a notable gap exists regarding the availability of datasets for other languages. For instance, the limited availability of high-quality synthetic speech methods for lower-resource languages hinders the development of robust antispoofing systems. As an example, the ThaiSpoof dataset, while a valuable contribution, is still in its early stages and lacks diverse spoofing attacks [15].

Indonesian, which is a language with significant diversity and unique phonetic characteristics, presents specific challenges for spoofing detection. To address this gap, we conduct a study to develop a comprehensive Indonesian-language antispoofing dataset. Recent findings have suggested that improving the quality and diversity of datasets is crucial for mitigating overfitting and ensuring robust model performance [4]. By creating a robust Indonesian-language dataset, we aim to facilitate the development of effective antispoofing systems for Indonesian and potentially other low-resource languages.

3 InaSpoof Dataset

Figure 1 shows the experimental pipeline of this study, which starts with data collection and curation. Data curation plays a pivotal role in the development of effective language technologies, especially for languages such as Indonesian with unique linguistic characteristics. Accurate and representative datasets

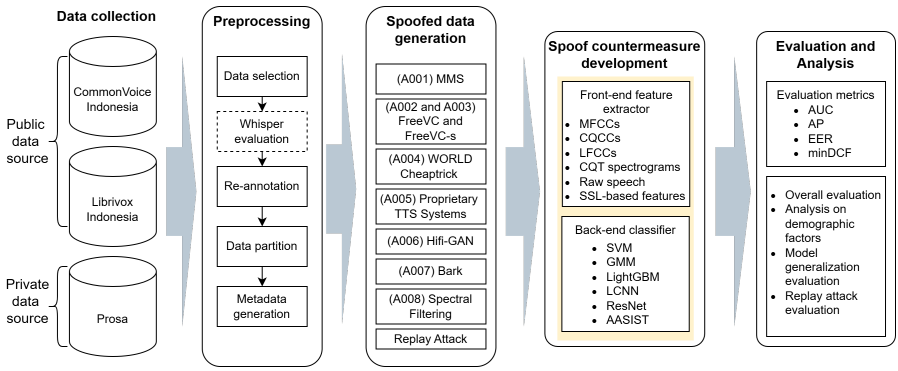

Figure 1: Experimental pipeline of InaSAS

Table 1: A brief summary of the InaSpoof-v1 dataset.

Subset	# Speakers	# Utterances	
		Bona fide	Spoofed
Train	55	2,360	41,292
Dev	90	2,002	34,699
Test	132	1,501	35,818

are essential for training machine learning models that can understand and process the nuances of Indonesian language. Carefully performing data curation involves tasks such as data collection, cleaning, annotation, and quality control. By ensuring the reliability and consistency of the curated data, researchers can build more robust and accurate spoofing detection models.

In this study, we carefully curated Indonesian speech recordings to construct our dataset, namely, the InaSpoof version 1 (InaSpoof-v1) dataset. This dataset was collected from three sources: CommonVoice, Librivox, and Prosa. Bona fide data were carefully selected from those three sources, ensuring diverse ranges of speakers and recording conditions. To obtain spoofing data, we employed various synthesis techniques, including speech vocoders, text-to-speech (TTS) models, and VC algorithms. Table 1 shows a summary of the InaSpoof-v1 dataset.

3.1 Bona Fide Data

After performing data collection, we conducted data cleaning and reannotation, followed by partitioning the data into training, development, and testing sets. The curated dataset adheres to the following criteria: (1) speech utterance durations between 3 and 8 seconds, (2) correct language usage, (3) exclusion of offensive or inappropriate words, and (4) adherence to standard Indonesian dialects.

To ensure a robust evaluation, we employed a stratified data splitting approach based on speaker IDs and recording conditions. This strategy helped maintain balanced speaker characteristic and recording environment distributions across the subsets. The test set is entirely independent, with none of its speakers present in the training data. The development set includes both overlapping and nonoverlapping speakers derived from the training data, allowing for a more comprehensive evaluation of the generalization capabilities of the tested model. To maintain sufficient training data for each speaker, we required at least five utterances per speaker in the training set. This approach helped prevent overfitting and ensured that the model could effectively learn from a diverse range of speaker variations.

3.1.1 Common Voice Dataset

Common Voice[1] is a crowdsourced dataset that aims to make voice recognition technology more accessible and inclusive. It consists of millions of short audio clips, each containing spoken sentences, in over 60 languages. Volunteers contribute to recording clips and reviewing others' work, helping to construct a diverse and high-quality dataset for training speech recognition systems. This open-source initiative contributes to democratizing voice technology by offering researchers and developers free access to data.

The Common Voice dataset for the Indonesian language, Version 16.1 (cv-corpus-16.1-2023-12-06-id), comprises approximately 57,614 utterances, totaling approximately 1.5 GB of data. The dataset includes labels, such as client IDs, paths, sentences (transcripts), upvotes, downvotes, and locales (IDs). Additionally, it includes incomplete labels such as ages, genders, accents, variants, and segments.

As the first step of the selection process, we analyzed the training, development, and testing subsets. We excluded the data that were not within the duration range that we had set. Since the original sampling frequency of this dataset was 32 kHz, we normalized it to 16 kHz. Next, we checked the word error rate (WER) by using Whisper Indonesia[2] (a fine-tuned OpenAI/Whisper-Medium model for version 11.0 of Indonesian CommonVoice, magic data, TITML, and the Google fleurs dataset). We selected only samples with WERs that were less than 5%.

After completing the initial preprocessing steps, we obtained a dataset consisting of 3055 training samples, 780 development samples, and 705 test samples. We named this subset CommonVoice-v0. Despite our selection efforts, the data still exhibited imbalance and included unknown labels. Additionally, many speakers had only one utterance, particularly in the test set. Owing to

[1] https://commonvoice.mozilla.org/en
[2] https://huggingface.co/cahya/whisper-medium-id

various factors, including the speaker distribution of each subset, this dataset might not have been ideal for conducting a cross-dataset evaluation [4].

To improve the quality of the dataset, we implemented additional processing steps on CommonVoice-v0. This included rebalancing the sample distribution across the training, development, and test sets and prioritizing more speakers in the training and development data. We removed outliers, including utterances derived from single speakers with more than 50 utterances and those with fewer than three occurrences. Additionally, speech segments exceeding eight seconds were excluded. The result was named the CommonVoice-v1 dataset. Table 2 summarizes the data distributions of CommonVoice-v0 and CommonVoice-v1.

Table 2: Distribution of the Common Voice dataset.

Version	Subset	# Utterances	# Speakers	Total duration (h)
v0	Train	3,074	4	5.55
	Dev	784	32	1.35
	Test	705	179	1.20
v1	Train	350	10	0.59
	Dev	330	49	0.55
	Test	438	110	0.71

3.1.2 Librivox Indonesia Dataset

The Librivox Indonesia[3] dataset, which was built from Librivox audiobooks, offers short snippets of Indonesian-language recordings. We specifically focused on the Indonesian languages contained within the Librivox collection. The original audiobooks varied greatly in length, but the speech clips included in this dataset were much shorter, ranging from just a few seconds to a maximum of 20 seconds each. The original sampling rate of this dataset was 44.1 kHz with stereo signals.

This dataset was built using multilingual forced alignment software. This software is versatile and works with various languages, including those with limited resources such as Acehnese, Balinese, and Minangkabau. Currently, the dataset comprises 8 hours of recordings across seven regional Indonesian languages: Acehnese, Balinese, Bugisnese, Indonesian, Minangkabau, Javanese, and Sundanese.

In this phase, we focused on the standard Indonesian language by using two audiobooks: "Mengelilingi Dunia dalam 80 Hari" (MD80H) and "Universal Declaration of Human Rights" (UDHR). Each audiobook was divided into training and test sets. Compared with the 15 samples of UDHR, MD80H has a more extensive test set with 588 samples. Similarly, the training set

[3] https://huggingface.co/datasets/indonesian-nlp/librivox-indonesia

for MD80H is significantly more extensive at 5,514 samples, whereas UDHR offers only 121 samples. Although these audiobooks are distinct, the same speaker narrated some utterances in both MD80H and UDHR.

The dataset exhibited a significant degree of imbalance, particularly in the number of samples per speaker ID, and some utterances were too short or too long. To address this issue, we rebalanced the sample distributions across the training, development, and test sets and removed outliers with more than 50 utterances per speaker. Table 3 summarizes the information of this dataset before (v0) and after conducting processing (v1).

Table 3: Distribution of the Librivox Indonesia dataset.

Version	Subset	# Utterances	# Speakers	Total duration (h)
v0	MD80H (Train)	5,514	9	6.10
	MD80H (Test)	588	9	0.65
	UDHR (Train)	121	1	0.20
	UDHR (Test)	15	1	0.03
v1	Train	1,250	6	1.40
	Dev	1,092	8	1.22
	Test	403	4	0.46

3.1.3 Prosa Dataset

The Prosa dataset provides a valuable resource for training speech processing models in Indonesian. These data were collected in the Prosa.ai[4] environment. This dataset, which was originally developed for training an Indonesian automatic speech recognition (ASR) system, consists of speech recordings captured in a controlled studio environment. The recordings feature clear audio due to the controlled setting and include content from formal meetings and news highlights.

Table 4 shows the distribution of the original recorded data (v0). Each subset encompasses 50 speakers with a balanced gender distribution (25 females and 25 males). Some speakers involved in the read and spontaneous sessions were identical, but not all of them. Each speaker contributed 20 utterances to each subset. While the utilized language was consistently Bahasa Indonesia, the dataset captures the diversity of Indonesian dialects by including recordings from eight major ethnicities: Javanese, Sundanese, Minang, Bataknese, Malay, Balinese, Sulawesi, and Papuan. This focus on dialectal variations makes the Prosa dataset particularly useful for tasks such as speaker identification or speech emotion recognition, where the background of the speaker may be relevant.

To establish a distinct test set for developing speech antispoofing systems, we carefully selected 18 speakers that were not represented in the training

[4] https://prosa.ai/

Table 4: Distribution of the Prosa dataset.

Version	Subset	# Utterances	# Speakers	Total duration (h)
v0	Read	1,000	50	1.43
	Spontan	1,000	50	2.33
v1	Train	760	39	1.40
	Dev	580	33	1.11
	Test	660	18	1.24

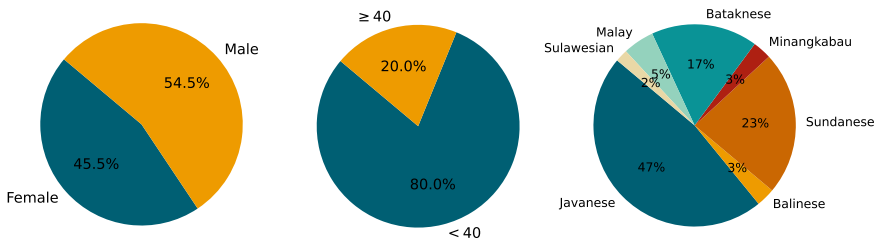

Figure 2: Demographic distribution of the InaSpoof-v1 dataset. Note that approximately half of the dataset lacks gender, age, or race information.

data. These speakers were evenly distributed across different recording sessions, ensuring a balanced evaluation. Table 4 summarizes the distribution of the Prosa dataset after completing the processing steps (v1). Furthermore, the demographic information of InaSpoof-v1 is outlined in Figure 2.

3.2 Spoof Data

Automatic speaker verification (ASV) systems leverage voice characteristics to identify authorized users. However, they are vulnerable to spoofing attacks [42]. In logical access attacks, attackers impersonate legitimate users with techniques such as TTS synthesis or VC. These synthesized or converted voices can trick the target ASV system into granting access to unauthorized individuals. Spoofing technology can even be so advanced that it can fool humans; these scenarios are sometimes referred to as voice cloning or voice deepfakes [49].

To date, research on ASV spoofing and deepfake countermeasures for Indonesian languages is scarce. Strong speech synthesis tools are essential for developing effective countermeasures. This study explored methods that could be used to generate high-quality spoofing data for the Indonesian language. We categorized five spoofing attacks utilizing the state-of-the-art multilingual TTS model (specifically, the massive multilingual speech (MMS) model [35]), VC (particularly FreeVC [29]), WORLD (particularly with the CheapTrick al-

gorithm) [33], and a collection of data from several proprietary TTS systems. Table 5 summarizes the generated spoofing data, including their distribution, based on their source datasets.

Table 5: A summary of the distributions of the bona fide and spoof data contained in InaSpoof-v1.

Subset	**Bona fide**	**Spoofed**							
		A001	**A002**	**A003**	**A004**	**A005**	**A006**	**A007**	**A008**
Train	2360	3744	17060	11920	2360	2,184	2360	760	2360
Dev	2002	3113	13960	10740	2002	900	2002	580	2002
Test	1501	3420	15420	11740	1381	585	1501	660	1501

3.2.1 MMS (A001)

TTS technology is evolving, moving beyond controlled environments and generating a wider variety of speech patterns; its ability to handle multiple languages is a key factor [8]. However, a major hurdle preventing the expansion of TTS to more languages is the limited amount of available training data, especially for those with fewer resources. One approach for addressing this data shortage involves the use of byte encoding to unify how text is represented and tested in English, Spanish, and Chinese [28]. Other studies have explored different ways to represent input text.

Recently, a significant advancement was achieved when a multilingual model developed by Facebook, namely, MMS [35], was created. This model can handle 1,406 languages and outperformed the existing solutions such as Whisper on a benchmark test with significantly less training data. The MMS TTS model is available here.[5]

To create our spoofing dataset, we leveraged the ability of the MMS model to generate speech in multiple languages. This approach was particularly advantageous because MMS supports several Indonesian accents. For the initial version of our dataset, we generated spoofed speech in three major Indonesian accents: Indonesian (ind), Javanese (jav), and Bataknese (bbc). The transcripts used for spoofed data generation purposes were derived from partial text transcriptions that were present in each real source dataset. We also generated approximately 100 sentences that are commonly spoken in daily life, each of which was approximately five seconds long. A current limitation of the MMS model is its restriction to generating speech from a single male speaker. The resulting spoofed speech signals are denoted as A001.

[5] https://huggingface.co/facebook/mms-tts

3.2.2 FreeVC (A002 and A003)

VC transfers a speaker identity, a prosody, and an emotion from a source to a target while preserving the content of the original voice. Typical VC systems employ separate models to convert acoustic features and generate waveforms from those features. However, these models are trained on different data (predicted vs. real speech), leading to a mismatch that reduces the quality of the final voice. ViTS [22], which is a one-stage model for both TTS and VC tasks, addresses this issue by connecting the conversion and waveform generation stages through a unique structure. This reduces the mismatch rate and improves the quality of the output. However, ViTS requires a text input and is limited to converting between predefined speakers.

We utilized FreeVC,[6] which is a text-free, one-shot VC system. It builds upon the ViTS architecture but eliminates the need for text annotations by learning to separate content information. FreeVC leverages WavLM [9] to extract speaker-independent features directly from waveforms. A bottleneck extractor then isolates the content information contained within these features. Additionally, spectrogram-resize (SR) data augmentation strengthens the ability of the constructed model to disentangle content by distorting speaker information while preserving content. To achieve one-shot conversion, a speaker encoder extracts speaker characteristics.

We evaluated two model variations: FreeVC, which uses a pretrained speaker encoder (A002), and FreeVC-s, which uses a nontrained speaker encoder (A003). The spoofed data were generated using the output of the MMS model, which incorporated speaker embeddings derived from approximately 100 randomly selected speakers within each subset of the bona fide dataset. This is why the total numbers of spoofed utterances obtained from A002 and A003 are much greater than that acquired from A001, which could only generate voices from one target speaker.

3.2.3 WORLD (A004)

WORLD[7] [33] represents a significant advancement in vocoder-based speech synthesis, offering a balance between high-quality outputs and real-time performance. Its modular design and efficient algorithms make it suitable for a wide range of speech technology applications. The synthesis process in WORLD involves the following steps: (1) generating source excitation based on F0 and aperiodicity information, (2) applying spectral envelope filtering to shape the excitation signal, and (3) producing the final speech waveform through an overlap-add synthesis process.

[6] https://github.com/OlaWod/FreeVC
[7] https://github.com/mmorise/World

To generate spoofed data, we modified the spectral envelope using CheapTrick [32]. CheapTrick is an accurate spectral envelope estimation algorithm based on a pitch-synchronous analysis. CheapTrick employs F0-based processing techniques, including adaptive windowing, spectral smoothing, and spectral recovery in the quefrency domain. We altered the pitch information by randomly increasing or decreasing it by 4–6 semitones.

3.2.4 Proprietary TTS systems (A005)

We leveraged several proprietary TTS systems beyond the state-of-the-art models to enrich the spoofed data with diverse synthesis styles. Owing to a lack of access to internal system details, including specific algorithms, training data, and architectures, a detailed, system-by-system analysis was infeasible. Consequently, we initially grouped these systems under a single "unknown attack" label (A005) to reflect this inherent uncertainty. The proprietary TTS systems included those developed by Azure,[8] Google,[9] Prosa.ai,[10] Elevenlabs,[11] Murf,[12] TTSMaker,[13] Narakeet,[14] and Play.HT[15].

We focused on generating Indonesian speech signals that were relevant to the banking transaction domain with various text contents. The generated audio consisted of extended utterances, averaging 30 seconds in length. These longer recordings were then segmented using Audacity[16] by following segmentation criteria that were consistent with the bona fide data described in Subsection 3.1.

3.2.5 Hifi-GAN (A006)

HiFi-GAN is a neural vocoder that is renowned for its ability to efficiently generate high-fidelity speech, making it a compelling choice for spoofed data generation in speech synthesis and VC research [24]. Its GAN architecture, comprising a generator and two discriminators (multiscale and multiperiod discriminators), allows it to learn complex mappings between mel-spectrogram inputs and raw waveforms. The key features that contribute to its performance include multiple receptive field fusion modules within the generator, enabling the model to capture diverse audio patterns, and a multiperiod dis-

[8] https://azure.microsoft.com/en-us/products/ai-services/ai-speech
[9] https://cloud.google.com/text-to-speech/docs/voice-types
[10] https://tts.prosa.ai/
[11] https://elevenlabs.io/text-to-speech
[12] https://murf.ai/
[13] https://ttsmaker.com/
[14] https://www.narakeet.com/languages/
[15] https://play.ht/text-to-speech/
[16] https://www.audacityteam.org/

criminator that was specifically designed to model and reproduce the crucial periodicities inherent in speech.

We selected HiFi-GAN because of its excellent balance between speech quality and generation speed. Its ability to synthesize high-fidelity audio is crucial for creating convincing spoofing attacks, which are essential for robustly training and evaluating speech antispoofing systems. Consistent with the process used for the A001 system, spoofing attack transcripts were derived from the partial text transcriptions provided with each real source dataset. Half of these generated utterances underwent randomized speed perturbations, with their durations varying by $10 - 20\%$.

3.2.6 Bark (A007)

Bark,[17] which is a transformer-based TTS model developed by Suno AI, was employed to generate spoofed speech because of its capacity to create highly realistic and diverse audio. Bark involves a similar approach to that of VALL-E [40]. The Bark architecture comprises four interconnected models: a semantic text model, a coarse acoustics model, a fine acoustics model, and an EnCodec decoder [10]. The semantic text model processes tokenized text, generating semantic tokens that capture the meaning of the text. These tokens are then input into the coarse acoustics model, which predicts the first two audio codebooks that are essential for EnCodec. The fine acoustics model, which is a noncausal autoencoder transformer, subsequently predicts the remaining codebooks on the basis of the previously generated codebooks. Finally, the EnCodec model synthesizes an audio waveform using the predicted codebooks. Crucially, the first three models can be conditioned on speaker embeddings, enabling the generation of speech with specific voice characteristics.

The speaker conditioning capabilities of Bark, combined with its ability to generate high-quality and diverse speech, made it a valuable tool for creating realistic spoofing attacks in our speech antispoofing research. As Bark does not yet support Indonesian, we utilized speaker conditioning by mapping Indonesian speaker identities to six proxy speakers (three per gender) provided by Suno AI. To minimize the distortion caused by noisy speech data (such as those found in Common Voice and LibriVox), we extracted Hubert features exclusively from the Prosa dataset, which was recorded in a clean environment. While the remaining components of the model were trained on English data, we anticipated that this approach could provide insights into the effectiveness of English-based pretrained models for Indonesian-language spoofing detection.

[17] https://github.com/suno-ai/bark

3.2.7 Spectral Filtering (A008)

A008 employs a transfer function-based VC system, which is similar to those used in the ASVspoof 2019 dataset generation process (A06 and A19) [42]. This system operates by analyzing the input voice signal using a source-filter model and then replacing the filters of the input signal with those of the target speaker. The modified filters are then used with the original residual signal to resynthesize the spoofed speech via a standard overlap-add technique. Because we directly used bona fide speech as our input, the text content remained unchanged.

4 Countermeasures for Spoofing Attacks

Spoofing detection is a critical technology in the fight against voice authentication fraud. It aims to distinguish between genuine speech derived from a human speaker (bona fide speech) and imitated/synthesized speech (spoofed speech). This is particularly important for systems that rely on voice biometrics, such as voice banking mechanisms or voice assistants.

Spoofing countermeasures typically consist of two primary components: frontend feature extraction and backend classification modules. Backend spoofing detection techniques analyze the speech characteristics extracted by the front end to identify inconsistencies or artifacts that are indicative of manipulation. These inconsistencies can arise from various spoofing methods, such as TTS or VC. More recently developed approaches utilize end-to-end systems that directly process raw speech to discriminate between bona fide and spoofed signals. One state-of-the-art end-to-end method is AASIST [18]. The following sections discuss frontend features, classifiers, and the end-to-end AASIST model employed for Indonesian spoofing detection.

4.1 Front-end Features

We extracted four features that are commonly used in speech antispoofing techniques: Mel-frequency cepstral coefficients (MFCCs) [36], constant-Q cepstral coefficients (CQCCs) [38], linear-frequency cepstral coefficients (LFCCs) [36], and spectrograms obtained from constant-Q transformations (CQT spectrograms) [50]. Figure 3 visualizes the features we used in this study.

4.1.1 MFCCs

MFCCs represent a powerful feature extraction technique that is widely used in speech and audio processing tasks. An MFCC offers a compressed representation of the spectral envelope of a sound signal, emphasizing the frequencies

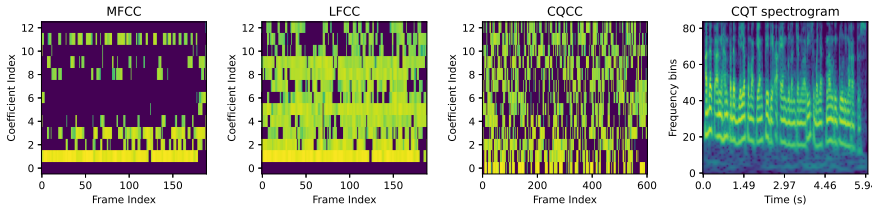

Figure 3: Frontend features utilized for our spoofing attack countermeasures.

that are perceived most distinctly by the human auditory system. This emphasis on pitch perception aligns well with the task of spoofing detection, where subtle variations in how genuine and spoofed voices distribute energy across different frequencies can be crucial for differentiating between them. By analyzing the extracted MFCC features, spoofing detection systems can effectively distinguish between bona fide speech and spoof speech.

4.1.2 CQCCs

The CQT [7] was reported to have high spoofing detection potential in speech analysis scenarios. Known for its effectiveness in various audio tasks, the CQT offers a high-resolution analysis process that is similar to that of wavelets but avoids the computational drawbacks of traditional wavelet techniques. CQCCs were proposed by combining the CQT with cepstral analysis. Since applying cepstral analysis directly to the CQT results in mismatched scales, Todisco et al. introduced a linearization step that allows for feature extraction to be properly performed by using the discrete cosine transform (DCT) [38]. This approach has the potential to identify subtle inconsistencies in spoofed speech that simpler methods might miss.

4.1.3 LFCCs

LFCCs offer an alternative approach to MFCCs for conducting feature extraction in speech processing tasks, including spoofing detection. An LFCC represents the spectral envelope of a sound signal on a linear frequency scale. This linearity can be advantageous in specific scenarios, particularly when the applied spoofing technique relies on manipulating specific frequency bands. For example, some VC methods might alter specific high-frequency regions of the target speech signal. By analyzing the corresponding LFCC, spoofing detection models can potentially identify these targeted manipulations in the frequency domain, even if they might be less noticeable at the mel-frequency scale used by MFCCs. While they may potentially be less aligned with human

auditory perception, LFCCs offer valuable complementary information to that of MFCCs and can be particularly useful when the nature of the employed spoofing techniques is not well defined.

4.1.4 CQT Spectrograms

The CQT offers a distinct advantage over the short-term Fourier transform (STFT) by providing a constant Q factor across the frequency spectrum. This leads to superior temporal resolutions at higher frequencies and better frequency resolutions at lower frequencies. Previous studies have highlighted the efficacy of the CQT in applications such as sound source separation and acoustic scene classification. Prior research has shown the potential of a CQT-based spectrogram for use in discriminating between spoofed and bona fide voices on the ASVspoof 2019 dataset, outperforming CQCCs [50].

4.1.5 Raw Speech

Using raw speech signals directly as features instead of traditional handcrafted features offers enhanced flexibility and improved performance across speech processing tasks, including speech spoof detection [19]. Modern deep learning architectures enable automatic extraction of complex, task-specific representations directly from raw waveforms, bypassing manual feature engineering limitations. Recent advancements demonstrate that raw waveform-based models achieve state-of-the-art results in multiple domains: speaker verification systems using raw inputs now achieve equal error rates below 1% on benchmark datasets, while end-to-end architectures show superior noise robustness in automatic speech recognition compared to conventional MFCC-based approaches [23].

4.1.6 SSL-based Features

Self-Supervised Learning (SSL) has revolutionized speech spoofing detection, overcoming limitations inherent in traditional handcrafted features like MFCCs. SSL models, such as wav2vec 2.0 [3] and XLS-R [5] learn hierarchical representations directly from raw audio waveforms. These representations capture both spectral and temporal characteristics, which are critical to distinguishing between bona fide and spoofed speech. This paradigm shift facilitates the development of end-to-end systems, bypassing the need for manual feature engineering and achieving state-of-the-art performance. Previous research suggested that SSL-based features exhibit robust generalization across a wide range of spoofing attack types, including artifacts introduced by TTS and VC techniques [37]. However, a significant challenge remains: SSL models often require considerable computational resources.

4.2 Backend Classifiers

Once the frontend features have been extracted, a backend classifier is employed to discriminate between the bona fide and spoofed speech signals. Various machine learning algorithms can be used for this task, each of which possesses its own strengths and weaknesses. Common choices include support vector machines (SVMs), Gaussian mixture models (GMMs), and deep neural networks (DNNs). In this study, we developed five classification models, including an SVM, a GMM, a light gradient boosting machine (LightGBM), a light convolutional neural network (LCNN), a residual network (ResNet), and an AASIST model.

4.2.1 SVMs

An SVM aims to find a hyperplane that optimally separates data points belonging to different classes; in this case, these classes bona fide and spoofed speech signals. The SVM algorithm works by mapping the input data (features extracted from speech signals) to a high-dimensional feature space and then finding the hyperplane with the maximum margin between the two classes. This margin maximization scheme helps improve the generalization performance of the SVM and mitigate overfitting. Several studies have demonstrated the effectiveness of SVMs in ASV-based spoofing detection tasks [36, 20, 31].

4.2.2 GMMs

GMMs form another popular technique for ASV-based spoofing detection scenarios. Numerous studies have validated their effectiveness, establishing them as common baseline models for this task [36, 42, 30, 31]. A GMM models the probability density function of a given dataset as a mixture of Gaussian distributions. By training separate GMMs for bona fide and spoofed speech signals, it is possible to classify new samples on the basis of their likelihood of belonging to either class. GMMs are particularly effective at modeling complex distributions that may not be well represented by a single Gaussian distribution. They can also capture the variability exhibited by speech signals, making them suitable for handling different speaker characteristics and recording conditions.

4.2.3 LightGBM

Compared with traditional gradient boosting algorithms, a LightGBM, which is a gradient boosting framework, introduces novel techniques such as histogram-based gradient boosting and exclusive feature bundling, significantly

reducing the required training time and memory usage demands [21]. These advantages make LightGBMs well suited for large-scale machine learning tasks. In the context of spoofed voice detection, a LightGBM can effectively model the complex relationships between features extracted from speech signals, enabling accurate classification. The ability of a LightGBM to handle large datasets and its fast training speed make it a promising choice for real-time voice spoofed detection applications. Despite their advantages, LightGBMs have been relatively unexplored as backend classifiers in most existing spoofed voice detection research. The effectiveness of a LightGBM for this task was investigated in this study.

4.2.4 LCNN

Convolutional neural networks (CNNs), which are known for their ability to recognize local patterns in data with minimal preprocessing steps, are commonly employed as DNN architectures for speech antispoofing [31]. The LCNN model has gained popularity in recent years because of its ability to achieve high performance while minimizing the incurred computational costs [26]. The LCNN model was initially proposed for replay attack detection [25]. In the ASVspoof 2019 challenge, it was enhanced with an angular margin-based softmax (A-softmax) activation function, which constrained the learned features to a unit hypersphere, improving the ability to discrimine between genuine and spoofed signals. This approach achieved impressive results, with equal error rates (EERs) of 1.86% and 0.54% in the LA and PA scenarios, respectively [26]. Given its promising performance, we adopted a CQT spectrogram as the frontend feature and the LCNN model as the backend classifier. The LCNN architecture follows the model derived from ASVspoof 2021 [30].

4.2.5 ResNet

A ResNet [16], which is a type of CNN, has been successfully applied to various tasks, including speech recognition and image classification. Its innovative design incorporates residual connections, enabling the network to learn residual functions and train deeper models without encountering the vanishing gradient problem. In the context of ASV-based spoofing detection, ResNets have demonstrated their ability to effectively extract discriminative features from speech signals, capturing complex patterns and relationships. ResNets have outperformed traditional machine learning methods and other DNN architectures in many ASV-based spoofing detection benchmarks. We adopted the ResNet architecture proposed in the ASVspoof 2019 challenge for our experiments [2].

4.2.6 AASIST

The AASIST model [18], which stands for audio antispoofing via integrated spectro-temporal graph attention networks, was designed to efficiently detect diverse spoofing attacks, eliminating the need for computationally intensive ensemble systems. AASIST processes raw waveform inputs, which are then encoded using a RawNet2-based encoder. It constructs and fuses spectral and temporal graphs into a unified spectro-temporal graph, integrating frequency- and time-related information. A key component is its novel heterogeneous stacking graph attention layer, which captures artifacts across various temporal and spectral intervals by using a specialized attention mechanism and a stack node. This architecture, which incorporates a novel graph maximization operation and a readout scheme, enabled AASIST to achieve a 20% relative improvement over the state-of-the-art models in ASVspoof 2019. To enhance performance, we integrated SSL-based features, leveraging their ability to capture robust, high-level representations from raw speech waveforms. Furthermore, we evaluated a lightweight variant, AASIST-L, on the InaSpoof-v1 dataset to assess its efficiency and effectiveness.

5 Experiments

This section describes the spoofing detection experiments conducted in this study. We first outline the hyperparameter tuning process. Next, we discuss the evaluation metrics used to assess the performance of the tested models. Owing to the novelty of the employed datasets for spoofing detection, we conducted a thorough analysis of each dataset individually and in combination with other datasets.

5.1 Hyperparameter Settings

Hyperparameters significantly influence the performance of machine learning models. While most of the hyperparameters contained in the tested feature extraction and machine learning models were set to their default values as used in the corresponding references, we explored alternative settings. Table 6 summarizes the feature extraction parameters, including MFCCs, LFCCs, CQCCs, and CQT spectrograms. For SSL-based features, we incorporated the wav2vec 2.0 XLS-R pre-trained model [5] with 300 million parameters.[18] Table 7 presents the classifier parameters used in our experiments, including those of the SVM, the GMM, the LightGBM, the LCNN, ResNet, and AASIST.

[18] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec/xlsr

Table 6: Feature extraction parameters. An asterisk (*) indicates that the default value used in the corresponding reference was employed.

Parameters	Feature			
	MFCC[19]	LFCC[20]	CQCC[21]	CQT spectrogram[22]
Sampling frequency	16 kHz	16 kHz	16 kHz	16 kHz
Frame length	30 ms	20 ms	*	*
Frame overlap	20 ms	10 ms	*	*
# Filters	20	20	*	*
Pre-emphasis coefficient	0.97	0.97	0.97	0.97
Window function	Hamming	Hamming	Hann	Hann
# Coefficients	13	20	20	-
Lag of the delta function	3	3	3	-
# Bins per octave	-	-	96	*
# Uniform samples in the first octave	-	-	20	*
Maximum length	-	-	-	200

Table 7: Machine learning model parameters. Values not listed in the table were set to their default values.

Parameters	Classifier					
	SVM	GMM	LightGBM	LCNN	ResNet	AASIST
Model	SVC[23]	GMM[24]	LGBM[25]	LCNN[26]	ResNet34[27]	AASIST, AASIST-L[28]
Kernel function	Poly	-	-	-	-	-
# Mixtures	-	2	-	-	-	-
Alpha	-	1	-	-	-	-
# Epochs	-	-	-	20	20	20
Batch size	-	-	-	128	128	4
Learning rate	-	-	-	0.0001	0.0001	0.0001
Early stopping patience	-	-	-	5	5	-
Optimizer	-	-	-	Adam	Adam	Adam
Loss function	-	-	-	Sparse categorical cross-entropy loss	Binary cross-entropy loss	Categorical cross-entropy loss

5.2 Evaluation Metrics

Our evaluation focused on a standalone spoofing detection scenario, similar to the deepfake task in ASVspoof 2021 [30], without incorporating automatic speaker verification. To evaluate the tested spoofing countermeasures, we utilized four metrics:

(i) **Area under the curve (AUC)**

The AUC is a commonly used metric for evaluating binary classification models. It measures the ability of a model to discriminate between positive and

[19] https://www.mathworks.com/help/audio/ref/mfcc.html
[20] https://github.com/smileslab/Comparative-Analysis-Voice-Spoofing
[21] https://github.com/asvspoof-challenge/2021/tree/main/LA/Baseline-CQCC-GMM/matlab/CQCC
[22] https://librosa.org/doc/main/generated/librosa.cqt.html
[23] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[24] https://github.com/jiwidi/gmm-classifier
[25] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html
[26] https://github.com/asvspoof-challenge/2021/tree/main/LA/Baseline-LFCC-LCNN
[27] https://github.com/nesl/asvspoof2019
[28] https://github.com/clovaai/aasist

negative instances. A receiver operating characteristic (ROC) curves plot the true-positive rate against the false-positive rate at different classification thresholds. The AUC represents the area under the ROC curve. A higher AUC indicates better overall classification performance, as the tested model can effectively distinguish between positive and negative instances across various classification thresholds.

(ii) **Average precision (AP)**

AP is a valuable metric for evaluating the performance of spoofing detection systems. It provides a measure of the ability of the tested model to retrieve relevant spoofed utterances while minimizing the number of false alarms. The AP is calculated by averaging the precision values attained at various recall levels. A higher AP value indicates that the associated model can effectively identify spoofed utterances with high precision, even at low recall levels. This is crucial in real-world applications where minimizing the number of false alarms is a priority. By using AP as an evaluation metric, researchers can assess the overall effectiveness of their spoofing detection systems and identify areas for improvement.

(iii) **Minimum detection cost function (minDCF)**

The minDCF was the primary metric used in Track 1 of the ASVspoof 5 challenge for evaluating spoofing detection systems [11]. It is a normalized detection cost function that accounts for the costs of false rejections and false alarms, as well as the prior probability of a spoofing attack. The minDCF is calculated by minimizing the normalized detection cost function (DCF') over the detection threshold (τ_{CM}). DCF' includes the false rejection rate, false-alarm rate, and cost ratio (β), the latter of which reflects the relative costs of false rejections and false alarms. The minDCF was compared with the actual detection cost function (actDCF) to assess the performance of the developed system under real-world conditions.

(iv) **Equal error rate (EER)**

The EER represents the point at which the false acceptance rate (FAR) and false rejection rate (FRR) are equal. In other words, the EER indicates the rate at which the tested model misclassifies both genuine and spoofed utterances equally. A lower EER indicates better overall performance, as the associated model can effectively discriminate between the two classes while minimizing the induced errors.

Considering the focus of this work on low-resource-language datasets, we acknowledge that training times might be of interest to some readers. We also report the average training times required by the different models to provide a more comprehensive picture of their efficiency.

5.3 Comparative Preliminary Analysis: InaSpoof-v0 and -v1

To evaluate the quality of the original dataset (the v0 version), we conducted preliminary experiments via nondeep learning methods: a LightGBM, an

SVM, and a GMM. These models were selected to reduce the required computational time and focus on the underlying data characteristics.

The results, as summarized in Table 8, indicate a high likelihood of overfitting. The models achieved exceptional performance on each dataset, with metrics such as their AUC and AP values approaching 1. Additionally, the minDCF and EER were also near 0. Previous research conducted using the Prosa and CommonVoice datasets reported poor cross-set evaluation performance [4]. This suggests that the distributions of these datasets may be imbalanced or that the performed task was overly simplistic, leading to the models easily learning the training set patterns without generalizing well to unseen data.

Table 8: Results of a performance comparison between InaSpoof-v0 and -v1. The comparison was conducted using a consistent set of spoofing attacks (A001-A004). Arrow directions indicate superior performance: ↑ denotes that higher values are better, ↓ signifies that lower values are better.

Feature	**Classifier**	**AUC (↑)**		**AP (↑)**		**minDCF (↓)**		**EER (%) (↓)**	
		v0	**v1**	**v0**	**v1**	**v0**	**v1**	**v0**	**v1**
MFCC	LightGBM	0.9763	0.7681	0.9453	0.5310	0.0053	0.1374	0.3000	6.0212
	SVM	0.7323	0.6182	0.4605	0.2711	0.8028	0.9010	53.5100	76.3271
	GMM	0.9595	0.8472	0.7427	0.5297	0.1497	0.5654	7.7600	28.8746
CQCC	LightGBM	0.9870	0.8766	0.9727	0.7315	0.0040	0.0685	0.1800	3.4011
	SVM	0.8397	0.6290	0.6847	0.2953	0.6093	0.8097	32.0600	74.1938
	GMM	0.9814	0.8863	0.7573	0.6250	0.0655	0.4225	3.5200	21.6942
LFCC	LightGBM	0.9795	0.9039	0.9571	0.7967	0.0076	0.0481	0.3800	2.0749
	SVM	0.7038	0.6008	0.4109	0.2459	0.7211	0.9198	59.2200	79.8290
	GMM	0.9915	0.9266	0.6985	0.7741	0.0245	0.2752	1.2025	14.2662

To address the overfitting issues observed in the v0 dataset, we implemented a refined data partitioning strategy for InaSpoof-v1. This comparison allowed us to assess the increased difficulty of the newer dataset. We evaluated the performance of the countermeasure models as standalone deepfake detection systems on each source dataset. Our findings confirm that InaSpoof-v1 presents a significantly greater challenge than v0 does. Table 8 illustrates this increased difficulty, as all the metrics decreased despite the inclusion of similar attack types.

Notably, SVM-based classifiers frequently misclassified the evaluation samples as spoofs on InaSpoof-v1, resulting in AUC scores near 0.6, AP values below 0.3, minDCF scores approaching 1, and EERs exceeding 70%. In general, the performance of the SVM on InaSpoof-v1 was substantially worse than that on InaSpoof-v0, where some evaluation samples were still correctly classified. Consequently, we excluded the SVM from further analyses because of its consistently poor performance. Conversely, we observed a clear performance advantage for the LightGBM on InaSpoof-v0, where it consistently achieved the highest scores.

5.4 Results Obtained on the InaSpoof-v1 Dataset

This analysis was aimed at investigating the effectiveness of the existing countermeasure models in terms of detecting Indonesian-language spoofing attacks, particularly when faced with the increased difficulty of the InaSpoof-v1 dataset. To achieve this goal, we conducted several experiments.

- First, we combined all the source datasets to create a more diverse and challenging dataset. Our initial evaluation aimed to assess the performance of the existing countermeasures on the InaSpoof-v1 dataset, specifically analyzing the attack types that rendered the system most vulnerable.
- Second, we conducted a demographic analysis of the spoofing detection performance of the tested methods, focusing on factors such as gender and dialect.
- Third, we performed a model generalization evaluation on private data. We trained and developed models on publicly available datasets and tested them on private, unseen data.
- Finally, we simulated real-world spoofing attacks using a subset of the InaSpoof-v1 dataset to evaluate the effectiveness of the model in terms of handling replay attacks. This experiment provided valuable insights into the performance of the models in a more realistic scenario.

5.4.1 Overall Evaluation

To comprehensively evaluate the robustness of different countermeasures against the InaSpoof-v1 dataset, we conducted an overall performance assessment by combining all the source datasets, creating a diverse and challenging evaluation environment. Our initial objective was to determine the effectiveness of the existing antispoofing techniques and identify the attack types that posed the greatest vulnerabilities. This evaluation encompassed wide ranges of features and classifiers, with the performance of the models assessed using standard metrics: the AUC, AP, minDCF, and EER. The detailed results are presented in Table 9.

This comprehensive evaluation demonstrated that the AASIST-based methods achieved the highest performance across most metrics, exhibiting superior AUC, AP, and minDCF scores. The combination of SSL-based features with the AASIST model yielded exceptional results, achieving an EER of less than 1% minDCF approaching 0. However, it is crucial to acknowledge that the use of pre-trained SSL-based features, which leverage information external to the training data, introduces a potential privacy concerns [39], preventing a direct, equitable comparison with other methods.

Table 9: Overall evaluation results obtained on InaSpoof-v1 (attacks detailed in Table 5). The top-performing countermeasure model utilizing features solely from the training dataset is indicated in black, while the top-performing model incorporating SSL-based features is indicated in blue.

Feature	**Classifier**	**AUC (↑)**	**AP (↑)**	**minDCF (↓)**	**EER (%) (↓)**
MFCC	LightGBM	0.6949	0.3761	0.1851	7.4622
	GMM	0.8385	0.1394	0.4292	20.5158
CQCC	LightGBM	0.8250	0.6291	0.1159	4.4514
	GMM	0.8149	0.2036	0.6310	28.9803
LFCC	LightGBM	0.8354	0.6413	0.1164	4.4111
	GMM	0.8460	0.2731	0.5329	24.9840
CQT	LightGBM	0.6815	0.3467	0.2097	8.4463
	LCNN	0.8512	0.6059	0.1891	6.5922
	ResNet	0.8209	0.6054	0.2367	8.4463
Raw speech	AASIST	0.9924	0.8703	0.1061	4.2051
	AASIST-L	**0.9928**	**0.9100**	**0.1052**	**4.1925**
SSL	AASIST	**0.9994**	**0.9888**	**0.0244**	**0.8658**

When utilizing raw speech as front-end input, AASIST-L, despite its reduced parameter count compared to AASIST, exhibited slightly improved performance in the overall InaSpoof-v1 evaluation. Nevertheless, this difference was not statistically significant ($p > 0.005$). Furthermore, LightGBM consistently achieved low EER values (below 5%) when simple features such as CQCCs and LFCCs were used, significantly outperforming the GMM-based approach.

Although the LightGBM generally yielded low EER values, its performance in terms of the AUC and AP metrics was considerably lower than that of the AASIST-based methods. This suggests that while the LightGBM excelled at balancing false positives and false negatives at a specific operating point (as reflected by the EER), it struggled to maintain high performance across the entire range of decision thresholds. This indicates a potential limitation in its ability to effectively discriminate between genuine and spoofed samples at varying confidence levels, particularly when compared with the more robust feature learning capabilities of AASIST.

Experiments conducted using CQT spectrograms with the LightGBM and CNN-based models (LCNN and ResNet) produced EER values within the range of 6–9%, highlighting the effectiveness of these feature-classifier combinations. This comprehensive evaluation provided a strong baseline for understanding the performance of various countermeasures on the challenging InaSpoof-v1 dataset.

Figure 4 presents the attack-specific performance of the top-performing methods identified in Table 9: CQT-LCNN, RawSpeech-AASIST, RawSpeech-AASIST-L, and SSL-AASIST. Each method displayed unique performance

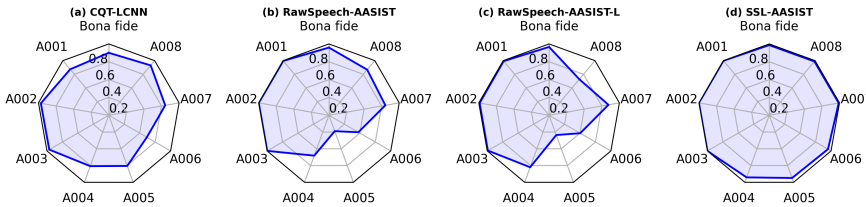

Figure 4: Accuracies achieved by the top 4 spoofing detection methods on the InaSpoof-v1 dataset divided by attack type. Methods: CQT-LCNN, RawSpeech-AASIST, and RawSpeech-AASIST-L.

patterns across the various attack types. To ensure a fair comparison, we analyze these results considering the use of features derived solely from the available training dataset versus those incorporating pre-trained SSL models.

Firstly, our experimental results suggest that CQT-LCNN struggled with vocoder-based attacks, particularly A006 (HiFi-GAN; accuracy of 69%) and A004 (CheapTrick; accuracy of 81%). However, it performed relatively well on the proprietary TTS-generated speech (A005; accuracy of 81%), outperforming the RawSpeech-AASIST-based methods in this category. Generally, CQT-LCNN achieved good accuracy ($\geq 85\%$) on the remaining attacks.

When employing raw speech waveforms as front-end features, both AASIST and AASIST-L demonstrated similar performance patterns in their respective radar plots. Although they achieved superior overall evaluation results in Table 9, both methods struggled to accurately detect the unknown proprietary TTS attack (A005; accuracy $< 50\%$) and vocoder-based attacks (A004 and A006). Notably, AASIST-L exhibited a 15% higher accuracy than AASIST for A004, but a 6% lower accuracy for A008. This discrepancy may be attributed to the disproportionately large number of spoofed utterances from attacks A001 to A003, compared to the limited data available for A004 to A008, despite the inclusion of more speaker embeddings. Consequently, the models struggled to recognize spoofed signals from these underrepresented attacks. The higher overall accuracy of AASIST models compared to LCNN model is likely due to the effective detection of bona fide speech signals, which, while fewer in number, are more easily distinguished.

Leveraging pre-trained SSL models, the AASIST architecture exhibited exceptional performance, achieving accuracies exceeding 95% for all attack types. This superior detection capability can be attributed to the inherent strengths of SSL representations. These models, trained on vast amounts of unlabeled audio data, learn hierarchical features that effectively encode both spectral and temporal information. Consequently, AASIST, when fed with

these powerful SSL features, is able to effectively distinguish between genuine and spoofed audio, even for highly challenging attack types.

In summary, CQT-LCNN and RawSpeech with AASIST-based models struggled with specific attack types, notably unknown TTS and vocoder-based attacks. Subsequently, utilizing pre-trained SSL models yielded exceptional performance, achieving detection error rates below 5% across all attack types.

Figure 5 illustrates the training and inference times required to build spoofing countermeasure models on the InaSpoof-v1 dataset using an RTX 2080 SUPER GPU. As depicted, the LightGBM exhibited the shortest training time, whereas the AASIST-based methods demonstrated the fastest inference times among those of the other methods. Although employing CQT spectrograms increased the dimensionality of the feature space, the corresponding training time increment induced for the LightGBM was relatively small, highlighting its efficiency. The LCNN generally required longer training times than ResNet did, but its inference time was slightly shorter. SSL-AASIST presented the highest computational demands, incurring significant training and inference costs. This is attributed to the inherent complexity of processing high-dimensional SSL features and the computational intensity of the graph-based operations within the AASIST model, highlighting the resource implications of employing SSL-driven architectures.

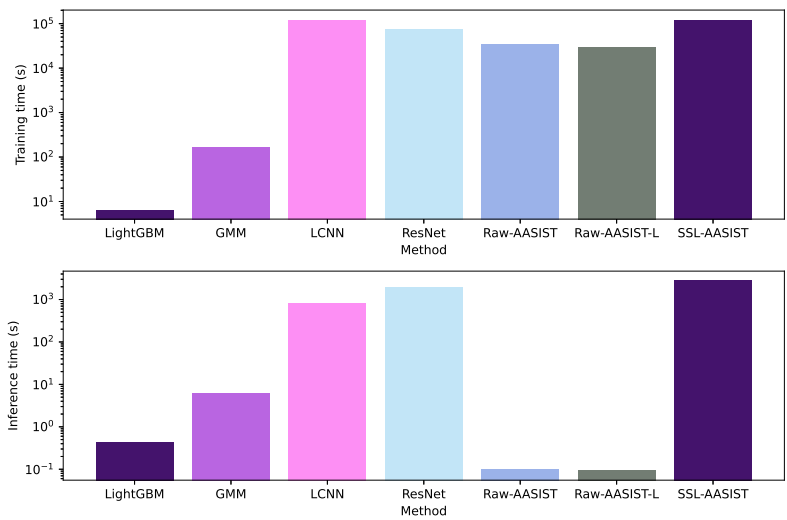

Figure 5: (Top) Training time (in seconds) required for each method on the InaSpoof-v1 dataset. (Bottom) Inference time (in seconds) required by each method for predicting the evaluation subset. The prefix 'Raw-' indicates the use of raw speech waveform features as input.

For applications demanding minimal inference latency, AASIST with raw speech waveform features emerged as the optimal choice, delivering high performance in distinguishing between bona fide and spoofed signals. Conversely, the LightGBM, despite its rapid training procedure, struggled to accurately identify bona fide signals that were not encountered during training. We observed that the presence of unseen speakers in the evaluation set led to a significant number of misclassifications, with real signals often being incorrectly identified as spoofs, resulting in an overall accuracy of approximately 0.67.

5.4.2 Analyzing the Influence of Demographics on Spoofing Detection Performance

A comprehensive evaluation of speech spoofing detection systems necessitates an understanding of the influence of demographic factors. Due to the inclusion of external resources in SSL-based features and the resulting near-perfect detection performance, analyzing demographic factors becomes challenging. Consequently, this section investigates the effects of gender and dialect variations on the performance of AASIST-L and CQT-LCNN, which exhibited the highest overall performance when utilizing only the training dataset features.

Figure 6 provides a detailed gender-specific analysis of the spoofing detection performance achieved on the InaSpoof-v1 dataset. Notably, AASIST-L demonstrated consistent performance across both male and female speakers, as reflected in its comparable minDCF and EER metrics. Conversely, CQT-LCNN revealed a significant performance discrepancy, with its minDCF and EER values approximately doubling for female speakers relative to those produced for male speakers. This suggests that CQT-LCNN encounters greater difficulty in terms of accurately identifying spoofed female speech. The high proportion of samples with unknown gender labels, particularly within the CommonVoice subset, likely contributed to an observed detection error increase.

Figure 7 examines the impact of the speaker age on the spoofing detection performance achieved within InaSpoof-v1. Both AASIST-L and CQT-LCNN exhibited declines in their spoofing detection accuracies for speakers aged 40 years and above. Although AASIST-L showed a relatively minor performance degradation relative to the results obtained for younger speakers, CQT-LCNN displayed a more significant divergence, as characterized by a 6% increase in its EER for the older age group.

Figure 8 provides an analysis of the spoofing detection performance achieved across various dialectal groups. Overall, AASIST-L and CQT-LCNN demonstrated consistent performance patterns across the major dialects: Javanese, Bataknese, and Unknown. However, the Melayu, Sulawesi, and Minang dialects, which are characterized by smaller sample sizes and the ab-

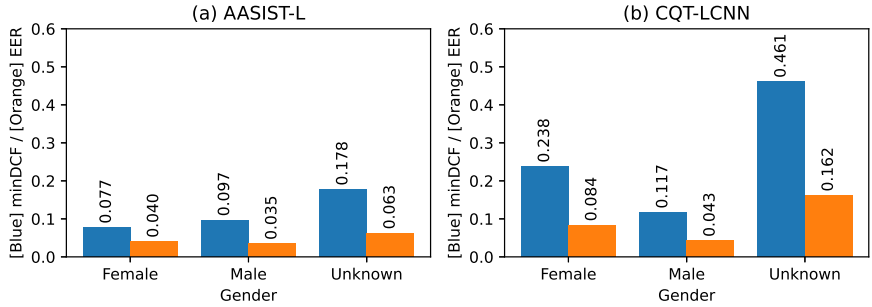

Figure 6: Analysis of the performance achieved across different genders. Abbreviation: Unk (Unknown).

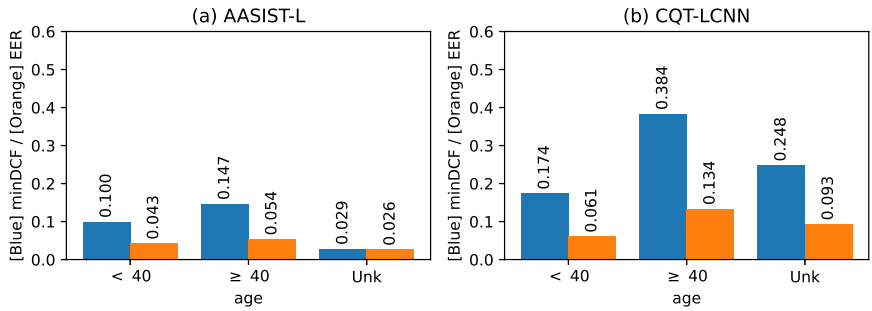

Figure 7: Analysis of the performance achieved across different ages. Abbreviation: Unk (Unknown).

sence of spoofed data in attacks A001, A002, and A003, exhibited a divergence in performance. Specifically, AASIST-L achieved near-perfect detection performance (with an EER approaching 0%), whereas the performance of CQT-LCNN varied, with an error rate ranging from 0–3%. Importantly, the MMS-generated dataset encompasses only the primary Indonesian dialects: standard Indonesian, Javanese, and Bataknese. The observed performance trends appear to have been influenced by the sample distribution. The Unknown dialect category, representing the largest sample size, presented the most significant challenge for accurately conducting spoofing detection. Similarly, the Javanese dialect, which was the second-most-populous category, also demonstrated increased prediction difficulty. These findings indicate that demographic imbalances lead to increased difficulty in detecting spoofing in dialects with higher attack variation compared to those with limited sample sizes.

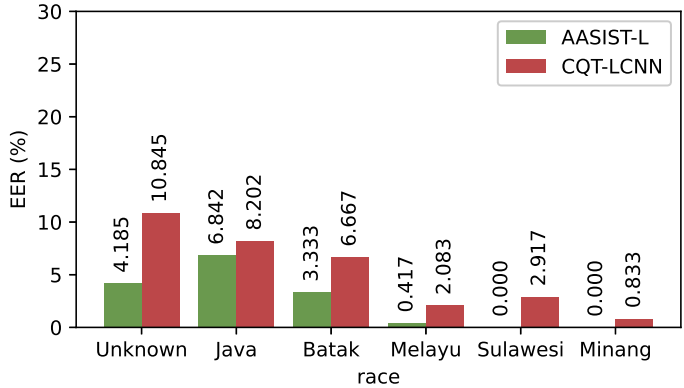

Figure 8: Analysis of the performance achieved across different races/dialects. The bona fide data included various Indonesian dialects, while the spoofed data primarily consisted of standard Indonesian with some Javanese and Bataknese accents. Abbreviations: Jav (Javanese), Btk (Bataknese), Mly (Malay/Betawi), Slw (Sulawesian), Mng (Minangkabau), Ind (Indonesian).

A detailed analysis of the bona fide and spoofed detection results produced by CQT-LCNN across different language groups reveals notable variations. For the bona fide data, which included diverse Indonesian dialects (Javanese, Bataknese, Malay, Sulawesian, and Minangkabau), the model performance exhibited significant disparities. Specifically, Bataknese, Malay, and Minangkabau were more accurately classified as bona fide data, whereas Sulawesian and Javanese presented greater classification challenges. Bataknese, Malay, and Minangkabau share linguistic similarities that are rooted in Austronesian languages, along with geographical and cultural connections. Conversely, Javanese and Sulawesian, despite also being Austronesian, display distinct linguistic and cultural characteristics. These phonetic and phonological feature variations likely contributed to the different performances attained by the model across various dialects.

5.4.3 Model Generalization Evaluation Conducted on a Private Dataset

We evaluated the generalizability of the tested models to unseen private data through an open-set evaluation, and the results are detailed in Table 10. The models were trained and developed on the publicly available CommonVoice and LibriVox datasets and subsequently evaluated on the Prosa dataset. Overall, the CQT-LCNN method demonstrated robust performance in terms of handling diverse speech synthesis attacks, achieving significantly lower minimum DCF and EER scores than the other methods did. The CQT-LCNN

Table 10: The results of the model generalization evaluation. The training and development sets were drawn from publicly available datasets (CommonVoice and Librivox), while the testing set was sourced from the Prosa dataset. The top-performing countermeasure model utilizing features solely from the training dataset is indicated in black, while the top-performing model incorporating SSL-based features is indicated in blue.

Feature	Classifier	AUC (↑)	AP (↑)	minDCF (↓)	EER (%) (↓)
MFCC	LightGBM	0.5319	0.0654	0.5476	24.2181
	GMM	0.7163	0.0976	0.8128	29.4269
CQCC	LightGBM	0.5159	0.0668	0.5326	19.6964
	GMM	0.5021	0.0505	1.0000	98.9375
LFCC	LightGBM	0.5287	0.0856	0.5354	21.6657
	GMM	0.7521	0.1276	0.6820	28.9260
CQT	LightGBM	0.5006	0.0502	0.5246	28.9075
	LCNN	0.7840	**0.4488**	**0.3466**	**12.5744**
	ResNet	0.6052	0.1617	0.6621	24.2581
Raw speech	AASIST	**0.8149**	0.2036	0.6310	28.9803
	AASIST-L	0.7269	0.1199	0.8213	32.8780
SSL	AASIST	**0.9984**	**0.9756**	**0.0346**	**1.6657**

method yielded the best scores for all the metrics, but its AUC values remained approximately 0.78, which were slightly lower than those of AASIST, suggesting limited discriminative capabilities in this open-set scenario. The LightGBM and GMM methods exhibited near-zero AP values, indicating a complete failure to generalize.

The generalization performance of the CNN-based methods revealed that the LCNN surpassed ResNet. The LCNN achieved an approximately 32% reduction in the minDCF and a 12% decrease in the EER, which was consistent with the overall evaluation findings presented in Table 9. However, the model generalization evaluation on private dataset yielded a contrasting result for the AASIST models using raw speech features (RawSpeech-AASIST). RawSpeech-AASIST consistently outperformed RawSpeech-AASIST-L across all evaluated metrics, demonstrating an approximate 0.2 improvement in minDCF and a 4% reduction in EER. This indicates that increased model capacity is essential for achieving robust performance on unseen data. The incorporation of SSL-based features yielded robust performance, even on unseen private data. While the generalization evaluation exhibited lower performance compared to the overall evaluation on known datasets presented in Table 9, the EER remained notably low, consistently below 5%.

Figure 9 presents the results of the model generalization evaluation, broken down by attack type. As anticipated, significant increases in the numbers of misclassifications were observed in this open-set scenario. CQT-LCNN exhibited the poorest performance on attack A006, achieving an accuracy of only approximately 50%. This method also struggled with attack A007, yielding an accuracy of just 51%. Attack A005 produced an accuracy of

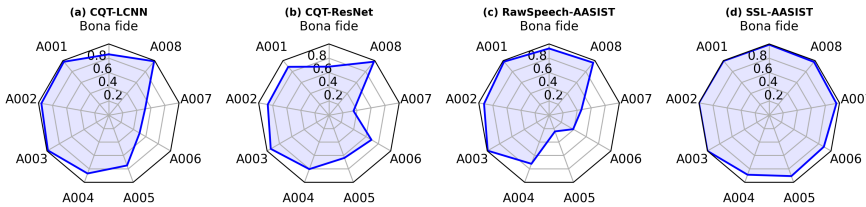

Figure 9: Accuracies achieved by the representative countermeasures in the model generalization evaluation, divided by attack type. Methods: CQT-LCNN, CQT-ResNet, RawSpeech-AASIST, and SSL-AASIST.

approximately 75%. It reached near-perfect accuracy (approaching 100%) on attacks A001, A002, and A008, whereas its accuracy was approximately 85% on bona fide data and attack A004. The performance of CQT-ResNet was worse than that of CQT-LCNN for almost all attack types except A006.

Building upon the attack-specific generalization results presented in Figure 9, we observed notable performance variations among the different methods. Although the AASIST models share a similar architecture, they exhibited distinct vulnerabilities to various attack types depending on the input features used. Both models effectively distinguished bona fide samples and attacks A001, A002, A003, and A008. However, the RawSpeech-AASIST model struggled with attacks A005 and A007, achieving accuracies below 50%, while the SSL-AASIST model achieved accuracies around 92%. Finally, the performance of RawSpeech-AASIST for vocoder-based attacks A004 and A006 remained consistently low, whereas the SSL-AASIST model achieved approximately 88% accuracy.

In conclusion, SSL-AASIST emerged as the optimal choice for robustly performing spoofing detection in open-set scenarios, where models must effectively generalize to unseen data and challenging recording environments.

5.5 Toward Actual Spoofing Through Replay Attack Simulation

5.5.1 Replay Attack Simulation

To simulate real-world spoofing attacks, we employed a straightforward approach involving the use of a smartphone, a laptop, an IC recorder, and two condenser microphones. This setup mimicked the potential vulnerabilities exhibited by ASV systems, where malicious actors could exploit playback devices to bypass security measures. By capturing audio samples from legitimate users and replaying them through the smartphone, laptop, IC recorder, and condenser microphones, we aimed to generate spoofed audio that could de-

ceive an ASV system. This approach allowed us to assess the effectiveness of our countermeasure models in terms of detecting and mitigating such attacks.

The simulation was conducted in two environments to mimic realistic conditions for replay attacks while maintaining controlled setups. The first environment was a personal apartment room with untreated acoustics and moderate ambient noise control, where the level of sound pressure of the background noise in the room was measured around 34 dB due to natural household sounds and mild noise control. Standard furniture created natural sound reflections and mild diffusion, and blankets were used to minimize the amount of reverberation. The second environment was a home recording room with better sound isolation and quieter surroundings. The sound pressure level of the background noise in the room was around 28 dB, offering a more controlled acoustic space for the replay attack data, incorporating a laptop and an IC recorder as recording devices.

Figure 10 shows the simplified process of our replay attack simulation. Table 11 lists the specifications of the devices used in the simulation. Initially, two condenser microphones were connected to a soundcard using XLR cables. The soundcard was then connected to a laptop via a soundcard cable. On the laptop, GarageBand was configured to receive two inputs: channel 1 for the attack microphone (AT2035) and channel 2 for the ASV system microphone (Ashley Studio Voice). The audio levels were adjusted to balance both inputs. The smartphone and laptop voice note applications were activated to prepare the microphones of both devices for recording. After the initial setup, the IC recorder and microphones were positioned 30 centimeters from the speaker. After this simulation, we obtained a replay attack dataset. All audio files were standardized by converting them to mono and 16-bit PCM. Table 12 summarizes the spoofing data generated from the replay attack simulation, including its distribution on the basis ofthe source datasets.

Table 11: Specifications of the devices employed in the replay attack simulation.

Device	Information
Condenser microphone (ASV)	Ashley Studio Voice Cardioid Condenser Microphone
Condenser microphone (Attacker)	AT2035 Cardioid Condenser Microphone
IC recorder	Sony ICD-UX71 Digital Voice Recorder
Laptop	MacBook Air (Mid-2013) built-in microphone with digital workstation audio software (GarageBand) installed
Smartphone	iPhone 8+ (built-in microphone)
Speaker	Bose Color Soundlink
Soundcard	Steinberg UR28M USB 2.0 Audio Interface

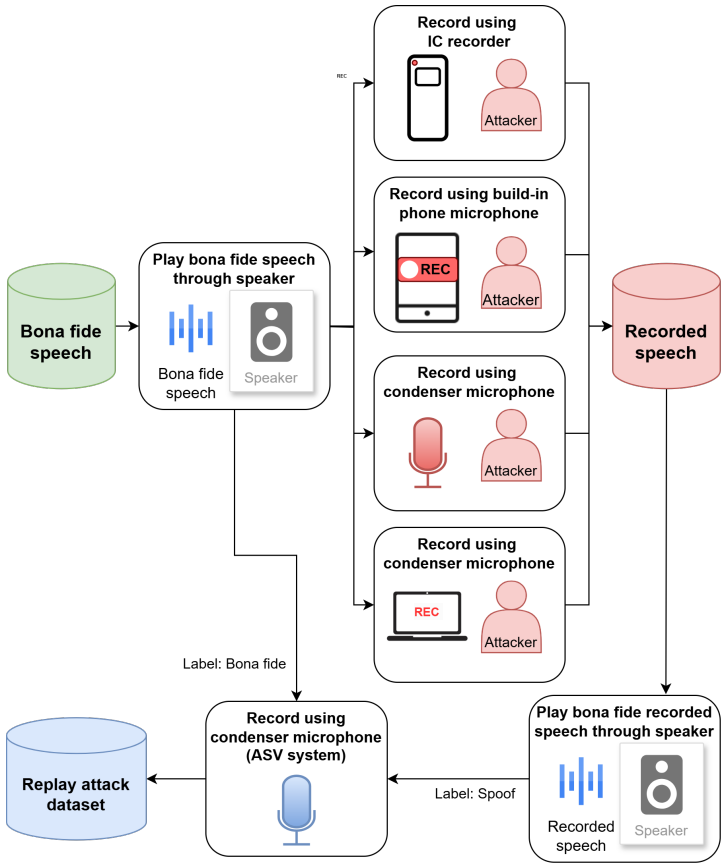

Figure 10: Replay attack simulation process.

Table 12: Distributions of the bona fide and spoofed samples contained in the replay attack dataset. Notes: HP = handphone microphone, Condenser = condenser microphone, IC = IC recorder, and Laptop = laptop microphone.

Source	Bona fide	Spoofed			
		HP	Condenser	IC	Laptop
CommonVoice	4,540	4,540	4,540	4,540	4,540
Prosa	2,000	2,000	2,000	2,000	2,000

5.5.2 Spoofing Countermeasures for Replay Attacks

We subsequently developed spoofing countermeasures for replay attack detection. The dataset was partitioned into training, development, and test sets at a 60:20:20 ratio and stratified by the speaker labels to maintain balance. Table

13 presents the results of the replay attack countermeasures. Across all classifiers, CQT spectrograms proved to be the most effective features. Notably, the LightGBM, LCNN, and ResNet models achieved minimal DCF values approaching zero and EER values below 5%, which significantly contrasted with the results observed in the speech synthesis countermeasure experiment (Subsection 5.4).

Table 13: Results of a simulated physical attack scenario where was is played back from a smartphone, an IC recorder, a laptop, and condenser microphones. The dataset used in this experiment was a balanced subset of CommonVoice and Prosa, matching the distribution of the bona fide data contained in InaSpoof-v1. Both playback types had equal representation. The best-performing methods are indicated in blue.

Feature	Classifier	AUC (↑)	AP (↑)	minDCF (↓)	EER (%) (↓)
MFCC	LightGBM	0.9207	0.8489	0.1331	5.5012
	GMM	0.9218	0.7620	0.2501	10.4217
CQCC	LightGBM	0.9061	0.8089	0.1506	6.0542
	GMM	0.7632	0.6287	0.8998	47.3596
LFCC	LightGBM	0.9301	0.8723	0.1296	4.9799
	GMM	0.8050	0.6942	0.7411	39.0061
CQT	LightGBM	0.9415	0.9006	0.0976	3.3841
	LCNN	0.9456	0.9001	0.0430	1.8478
	ResNet	**0.9882**	**0.9492**	**0.0204**	**1.1443**
Raw speech	AASIST	0.9442	0.8584	0.3534	13.2528
	AASIST-L	0.9332	0.8300	0.3927	14.3166

Regarding the performance of the tested classifiers, our evaluation demonstrated that the CQT-based models consistently outperformed those trained on MFCC, CQCC, and LFCC features. As shown in Table 13, the ResNet classifier achieved the highest AUC (0.9882) and AP (0.9492) values, along with the lowest minDCF (0.0204) and EER (1.1443%). These results indicate that, in our evaluation, the CNN-based classifiers, particularly ResNet, effectively differentiated replayed speech from bona fide speech. The LCNN model also exhibited comparable performance, reinforcing the efficacy of CNN-based methods for serving as replay attack countermeasures. Additionally, the LightGBM, while performing adequately, presented higher minDCF and EER values than the CNN-based methods did, suggesting limitations in terms of handling complex replay variations. The GMM-based classifiers, especially those with CQCC and LFCC features, performed significantly worse, indicating their ineffectiveness against these variations.

Surprisingly, the AASIST-based methods exhibited notably decreased effectiveness when applied to our replay attack scenario. This finding was particularly significant, as AASIST typically demonstrated strong performance in other spoofing detection tasks. Our hypothesis is that the architecture of AASIST, while highly effective for logical access attacks, struggles to discern the

subtle environmental variations introduced by replay mechanisms, even when the underlying voice characteristics remain similar. Importantly, AASIST was originally designed and optimized for logical access spoofing, specifically targeting the ASVspoof 2019 dataset [18]. Therefore, its limitations in terms of handling the unique challenges posed by replay attacks were not entirely unexpected but rather highlight the importance of considering the specific nature of the given spoofing attack when selecting and designing countermeasures.

We then investigated the impacts of replay device variations on the detection accuracies of the models, and the results are presented in Figure 11. Consistent with our observations in the synthesized speech experiments, the LightGBM demonstrated weakness in terms of detecting bona fide speech, particularly when encountering speakers that were unseen during training. Conversely, the CNN-based methods (CQT-LCNN and CQT-ResNet) exhibited remarkably robust performance, maintaining near-perfect accuracy (approximately 100%) across all playback sources, with minimal device-specific variations. Notably, the AASIST-based methods exhibited performance declines when presented with replays recorded using higher-quality devices, specifically laptop and condenser microphones, where their accuracies decreased to approximately $80 - 85\%$.

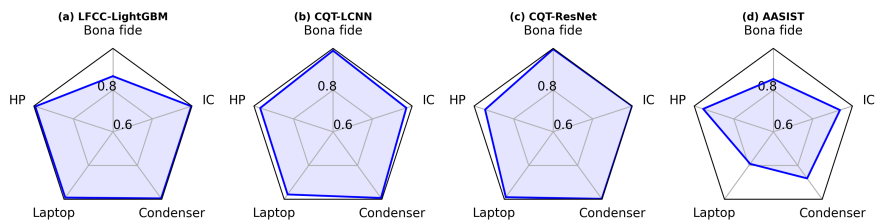

Figure 11: Accuracies achieved by the representative methods on the replay attack datasets produced by different recording devices. Methods: LFCC-LightGBM, CQT-LCNN, CQT-ResNet, and AASIST.

Finally, we evaluated the performance of the tested replay attack countermeasures under a cross-corpus setting. Given the composition of our dataset, which includes CommonVoice and Prosa, we conducted two experimental scenarios: training on CommonVoice and evaluating the models on Prosa, and vice versa. Table 14 presents the performance achieved by the representative countermeasures–CQT-LightGBM, CQT-LCNN, CQT-ResNet, and AASIST. These results highlight the significant challenge faced by countermeasures in terms of distinguishing between bona fide speech and replayed speech across different corpora.

Table 14: Cross-corpus evaluation results obtained on replay attack datasets.

Train-Eval Dataset	Countermeasure	AUC (↑)	AP (↑)	minDCF (↓)	EER (%) (↓)
CommonVoice-Prosa	CQT-LightGBM	0.5717	0.3473	0.5678	20.0000
	CQT-LCNN	0.5117	0.2559	0.6200	25.9948
	CQT-ResNet	0.6721	0.4208	0.4331	20.0000
	AASIST	0.6606	0.3949	0.8882	39.8385
Prosa-CommonVoice	CQT-LightGBM	0.7191	0.3124	0.3709	13.8830
	CQT-LCNN	0.6952	0.2908	0.3275	16.1702
	CQT-ResNet	0.7383	0.3234	0.3626	21.7021
	AASIST	0.7911	0.5248	0.6722	29.2199

Furthermore, the countermeasures trained on Prosa consistently outperformed those trained on CommonVoice, with the best countermeasure exhibiting a minimum DCF gap exceeding 0.1, despite Prosa having significantly fewer samples. This underscores the critical importance of high-quality data, and this finding is likely attributable to the studio-recorded nature of the Prosa dataset. In contrast, the CommonVoice dataset exhibits greater diversity but lacks controlled recording conditions, encompassing transcription, noise, and recording device variations.

While this setup provided a valuable initial step for understanding the challenges related to spoofing attacks, importantly, real-world scenarios may involve more sophisticated techniques and devices. Future research should explore more advanced spoofing methods and develop robust countermeasures to address these evolving threats.

6 Limitations and Challenges

While this study provides valuable insights into the challenges and potential solutions related to the development of Indonesian speech antispoofing systems, several limitations and areas for future work are identified.

1. **Language Coverage:** The current study focused solely on the Indonesian language, albeit encompassing major dialects. Expanding the research to other languages would provide a more comprehensive understanding of the challenges involved in cross-lingual spoofing detection.
2. **Integration with ASV Systems:** The conducted experiments have yet to integrate spoofing detection fully with ASV systems. To simulate real-world scenarios, future research should explore the impacts of spoofing attacks on the overall performance of ASV systems and develop strategies to mitigate these threats.
3. **Limited Spoofing Attack Diversity:** The study primarily considered five types of spoofing attacks. Real-world attacks can be more diverse and sophisticated, involving advanced techniques and various playback devices. Future

research should investigate a wider range of spoofing attacks, including those that exploit vulnerabilities in specific ASV systems.

4. **Controlled Recording Environments:** The audio data used in this study were collected in relatively controlled environments. Real-world scenarios often involve noisy and adverse acoustic conditions. Future research should explore the robustness of countermeasure models under noisy and reverberant conditions.
5. **Limited Exploration of Different Features and Classifiers:** This study employed limited sets of features and classifiers. A more comprehensive exploration of various feature engineering techniques and advanced machine learning models could improve the performance of spoofing detection systems. Additionally, hyperparameter tuning can further optimize the performance of these models.

We acknowledge the limitations of this study and recognize the importance of the identified areas for future work. By addressing these, we will contribute to the development of more robust and effective spoof detection systems for the Indonesian language and beyond. In particular, a comprehensive theoretical analysis, which was not feasible within the current scope, presents a significant opportunity for future research.

7 Conclusion and Future Work

This paper provides a comprehensive analysis of the challenges and opportunities related to Indonesian-language spoofing detection. Our findings highlight the significant challenge posed by the InaSpoof-v1 dataset. The increased diversity and complexity of spoofing attacks necessitate the development of robust and adaptive countermeasures.

While traditional machine learning methods such as the LightGBM proved effective in controlled environments (with known attacks, speakers, and environments), CNN-based and end-to-end approaches, notably ResNet and AASIST, demonstrated significantly superior performance in complex, real-world scenarios. Furthermore, the integration of SSL features, which learn robust representations from large amounts of unlabeled data, significantly enhances the performance of these end-to-end models, particularly in unseen and challenging conditions. Specifically, ResNet excelled as a replay attack countermeasure, whereas AASIST proved most effective against synthesized speech attacks in terms of overall performance.

Additionally, our demographic analysis revealed that the performance of the models varied across different dialects, with some dialects being more susceptible to spoofing attacks. These findings underscore the importance of considering demographic factors when developing effective spoofing detection

systems. To advance the field of spoofing detection, future research should prioritize the areas outlined in Section 6. By addressing these limitations, we can significantly increase the security of voice-based authentication systems and safeguard against a wide range of spoofing attacks.

Acknowledgments

This work was financially supported by PPMI ITB 2024, JSPS KAKENHI (25K21245), and JAIST Research Grant (Fundamental Research). The authors would also like to express their gratitude to Prosa.ai for partially providing their dataset. Additionally, this work was a part of the ASEAN IVO project titled "Spoof Detection for Automatic Speaker Verification"(www.nict.go.jp/en/asean_ivo).

References

[1] A. F. Aji, G. I. Winata, F. Koto, S. Cahyawijaya, A. Romadhony, R. Mahendra, K. Kurniawan, D. Moeljadi, R. E. Prasojo, T. Baldwin, J. H. Lau, and S. Ruder, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, ed. S. Muresan, P. Nakov, and A. Villavicencio, Association for Computational Linguistics, 2022, 7226–49, DOI: 10.18653/V1/2022.ACL-LONG.500.

[2] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep Residual Neural Networks for Audio Spoofing Detection", in *Proc. of Interspeech 2019, Graz, Austria, September 15-19, 2019*, ed. G. Kubin and Z. Kacic, ISCA, 2019, 1078–82, DOI: 10.21437/INTERSPEECH.2019-3174.

[3] A. Angra, H. Muralikrishna, D. A. Dinesh, and V. Thenkanidiyoor, "Exploring Aggregated wav2vec 2.0 Features and Dual-Stream TDNN for Efficient Spoken Dialect Identification", *IEEE Access*, 13, 2025, 3115–29, DOI: 10.1109/ACCESS.2024.3523951, https://doi.org/10.1109/ACCESS.2024.3523951.

[4] S. A. Arief, C. O. Mawalim, and D. P. Lestari, "Indonesian Speech Anti-Spoofing System: Data Creation and CNN Models", in *Proc. of ICAICTA, September 28-30, 2024, Singapore*, 2024.

[5] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale", in *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, ed. H. Ko and J. H. L. Hansen, ISCA, 2022, 2278–82, DOI: 10.21437/INTERSPEECH.2022-143, https://doi.org/10.21437/Interspeech.2022-143.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: a framework for self-supervised learning of speech representations", in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, *NIPS '20*, Vancouver, BC, Canada: Curran Associates Inc., 2020, ISBN: 9781713829546.

[7] J. Brown, "Calculation of a Constant Q Spectral Transform", *Journal of the Acoustical Society of America*, 89 (January), January 1991, 425–, DOI: 10.1121/1.400476.

[8] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone", in *Proc. of ICML*, Vol. 162, PMLR, July 2022, 2709–20, https://proceedings.mlr.press/v162/casanova22a.html.

[9] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing", *IEEE Journal of Selected Topics in Signal Processing*, 16(6), October 2022, 1505–18, ISSN: 1941-0484, DOI: 10.1109/jstsp.2022.3188113.

[10] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression", *arXiv preprint arXiv:2210.13438*, 2022.

[11] H. Delgado, N. Evans, J.-w. Jung, T. Kinnunen, I. Kukanov, K. A. Lee, X. Liu, H.-j. Shim, M. Sahidullah, H. Tak, M. Todisco, X. Wang, and J. Yamagishi, "ASVspoof 5 Evaluation Plan", *tech. rep.*, ASVspoof consortium, June 28, 2024, http://www.asvspoof.org/.

[12] H. Delgado, M. Todisco, M. Sahidullah, N. W. D. Evans, T. Kinnunen, K. Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements", in *Odyssey 2018: The Speaker and Language Recognition Workshop, 26-29 June 2018, Les Sables d'Olonne, France*, ed. A. Larcher and J. Bonastre, ISCA, 2018, 296–303, DOI: 10.21437/ODYSSEY.2018-42.

[13] S. K. Ergunay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing", in *IEEE 7th International Conference on Biometrics Theory, Applications and Sys-*

tems, BTAS 2015, Arlington, VA, USA, September 8-11, 2015, IEEE, 2015, 1–6, DOI: 10.1109/BTAS.2015.7358783.

[14] N. W. D. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification", in *Proc. of INTERSPEECH 2013, Lyon, France, August 25-29, 2013*, ed. F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, ISCA, 2013, 925–9, DOI: 10.21437/INTERSPEECH.2013-288.

[15] K. Galajit, T. Kosolsriwiwat, M. Unoki, C. O. Mawalim, P. Aimmanee, W. Kongprawechnon, W. P. Pa, A. Chaiwongyen, T. Racharak, S. Boonkla, H. Yassin, and J. Karnjana, "ThaiSpoof: A Database for Spoof Detection in Thai Language", in *2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2023, 1–6, DOI: 10.1109/iSAI-NLP60301.2023.10354956.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *Proc. of IEEE CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, 770–8, DOI: 10.1109/CVPR.2016.90.

[17] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units", *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29 (October), October 2021, 3451–60, ISSN: 2329-9290, DOI: 10.1109/TASLP.2021.3122291, https://doi.org/10.1109/TASLP.2021.3122291.

[18] J. Jung, H. Heo, H. Tak, H. Shim, J. S. Chung, B. Lee, H. Yu, and N. W. D. Evans, "AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks", in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, IEEE, 2022, 6367–71, DOI: 10.1109/ICASSP43922.2022.9747766.

[19] J. Jung, Y. J. Kim, H. Heo, B. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition", in *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, ed. H. Ko and J. H. L. Hansen, ISCA, 2022, 2228–32, DOI: 10.21437/INTERSPEECH.2022-126, https://doi.org/10.21437/Interspeech.2022-126.

[20] M. Kamble, H. Sailor, H. Patil, and H. Li, "Advances in anti-spoofing: from the perspective of ASVspoof challenges", *APSIPA Transactions on Signal and Information Processing*, 9 (January), January 2020, DOI: 10.1017/ATSIP.2019.21.

[21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", in *Proc. of Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, 3146–54.

[22] J. Kim, J. Kong, and J. Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech", 2021, arXiv: 2106.06103 [cs.SD].

[23] S. Kim, C. Lim, J. Heo, J. Kim, H. Shin, K. Koo, and H. Yu, "MR-RawNet: Speaker verification system with multiple temporal resolutions for variable duration utterances using raw waveforms", *CoRR*, abs/2406.07103, 2024, DOI: 10.48550/ARXIV.2406.07103, arXiv: 2406.07103, https://doi.org/10.48550/arXiv.2406.07103.

[24] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis", in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 2020, https://proceedings.neurips.cc/paper/2020/hash/c5d736809766d46260d816d8dbc9eb44-Abstract.html.

[25] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio Replay Attack Detection with Deep Learning Frameworks", in *Proc. of Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*, ISCA, 2017, 82–6, DOI: 10.21437/INTERSPEECH.2017-360.

[26] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge", in *Proc. of Interspeech 2019, Graz, Austria, September 15-19, 2019*, ISCA, 2019, 1033–7, DOI: 10.21437/INTERSPEECH.2019-1768.

[27] M. P. Lewis, "Ethnologue: Languages of the World (16th ed.)", Retrieved 27 October 2024, 2009, https://www.ethnologue.com/country/ID/.

[28] B. Li, Y. Zhang, T. Sainath, and W. Chan, "Bytes Are All You Need: End-to-end Multilingual Speech Recognition and Synthesis with Bytes", in *Proc. of ICASSP*, 2019, 5621–5, DOI: 10.1109/ICASSP.2019.8682674.

[29] J. Li, W. Tu, and L. Xiao, "Freevc: Towards High-Quality Text-Free One-Shot Voice Conversion", *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, 1–5.

[30] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. W. D. Evans, A. Nautsch, and K. A. Lee, "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild", *IEEE ACM Trans. Audio Speech Lang. Process.*, 31, 2023, 2507–22, DOI: 10.1109/TASLP.2023.3285283.

[31] A. Mittal and M. Dua, "Automatic speaker verification systems and spoof detection techniques: review and analysis", *Int. J. Speech Technol.*, 25(1), 2022, 105–34, DOI: 10.1007/S10772-021-09876-2.

[32] M. Morise, "CheapTrick, a spectral envelope estimator for high-quality speech synthesis", *Speech Commun.*, 67, 2015, 1–7, DOI: 10.1016/J.SPECOM.2014.09.003.

[33] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications", *IEICE Trans. Inf. Syst.*, 99-D(7), 2016, 1877–84, DOI: 10.1587/TRANSINF.2015EDP7457.

[34] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does Audio Deepfake Detection Generalize?", in *Proc. of Interspeech 2022, Incheon, Korea, September 18-22, 2022*, ed. H. Ko and J. H. L. Hansen, ISCA, 2022, 2783–7, DOI: 10.21437/INTERSPEECH.2022-108.

[35] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. M. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling Speech Technology to 1,000+ Languages", *ArXiv*, abs/2305.13516, 2023.

[36] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection", in *Proc. of INTERSPEECH 2015, Dresden, Germany, September 6-10, 2015*, ISCA, 2015, 2087–91, DOI: 10.21437/INTERSPEECH.2015-472.

[37] H. Tak, M. Todisco, X. Wang, J. Jung, J. Yamagishi, and N. W. D. Evans, "Automatic Speaker Verification Spoofing and Deepfake Detection Using Wav2vec 2.0 and Data Augmentation", in *Odyssey 2022: The Speaker and Language Recognition Workshop, 28 June - 1 July 2022, Beijing, China*, ed. T. F. Zheng, ISCA, 2022, 112–9, DOI: 10.21437/ODYSSEY.2022-16, https://doi.org/10.21437/Odyssey.2022-16.

[38] M. Todisco, H. Delgado, and N. W. D. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification", *Comput. Speech Lang.*, 45, 2017, 516–35, DOI: 10.1016/J.CSL.2017.01.001.

[39] W. Tseng, W.-T. Kao, and H.-y. Lee, "Membership Inference Attacks Against Self-supervised Speech Models", 2021, 5040–4, DOI: 10.21437/interspeech.2022-11245.

[40] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers", *CoRR*, abs/2301.02111, 2023, DOI: 10.48550/ARXIV.2301.02111, https://doi.org/10.48550/arXiv.2301.02111.

[41] X. Wang, H. Delgado, H. Tak, J. Jung, H. Shim, M. Todisco, I. Kukanov, X. Liu, M. Sahidullah, T. Kinnunen, N. W. D. Evans, K. A. Lee, and J. Yamagishi, "ASVspoof 5: Crowdsourced Speech Data, Deepfakes, and Adversarial Attacks at Scale", *CoRR*, abs/2408.08739, 2024, DOI: 10.48550/ARXIV.2408.08739.

[42] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. W. D. Evans, M. Sahidullah, V. Vestman, T. H. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, and Z. Ling, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech", *Comput. Speech Lang.*, 64, 2019, 101114.

[43] Z. Wu, N. W. D. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey", *Speech Commun.*, 66, 2015, 130–53, DOI: 10.1016/J.SPECOM.2014.10.005.

[44] Z. Wu, A. Khodabakhsh, C. Demiroglu, J. Yamagishi, D. Saito, T. Toda, and S. King, "SAS: A speaker verification spoofing database containing diverse attacks", in *Proc. of ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, IEEE, 2015, 4440–4, DOI: 10.1109/ICASSP.2015.7178810.

[45] Z. Wu, T. Kinnunen, N. W. D. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge", in *Proc. of INTERSPEECH 2015, Dresden, Germany, September 6-10, 2015*, ISCA, 2015, 2037–41, DOI: 10.21437/INTERSPEECH.2015-462.

[46] Y. Xiao and R. K. Das, "XLSR-Mamba: A Dual-Column Bidirectional State Space Model for Spoofing Attack Detection", *IEEE Signal Processing Letters*, 32, 2025, 1276–80, DOI: 10.1109/LSP.2025.3547861.

[47] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. W. D. Evans, and H. Delgado, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection", *CoRR*, abs/2109.00537, 2021, https://arxiv.org/abs/2109.00537.

[48] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, C. Fan, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, Z. Wen, and H. Li, "ADD 2022: the first Audio Deep Synthesis Detection Challenge", in *Proc. of ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, IEEE, 2022, 9216–20, DOI: 10.1109/ICASSP43922.2022.9746939.

[49] J. Yi, J. Tao, R. Fu, X. Yan, C. Wang, T. Wang, C. Y. Zhang, X. Zhang, Y. Zhao, Y. Ren, L. Xu, J. Zhou, H. Gu, Z. Wen, S. Liang, Z. Lian, S. Nie, and H. Li, "ADD 2023: the Second Audio Deepfake Detection Challenge", in *Proc. of the Workshop on Deepfake Audio Detection and Analysis co-located with (IJCAI 2023), Macao, China, August 19, 2023*, ed. J. Tao, H. Li, J. Yi, and C. Fan, Vol. 3597, *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, 125–30, https://ceur-ws.org/Vol-3597/paper21.pdf.

[50] P. A. Ziabary and H. Veisi, "A Countermeasure Based on CQT Spectrogram for Deepfake Speech Detection", in *Proc. of (ICSPIS)*, 2021, 1–5, DOI: 10.1109/ICSPIS54653.2021.9729387.