PROACTIVE DETECTION OF SPEAKER IDENTITY MANIPULATION WITH NEURAL WATERMARKING

Wanying Ge, Xin Wang, Junichi Yamagishi
National Institute of Informatics
Chiyoda-ku, Tokyo 101-8430, Japan
{gewanying,wangxin,jyamagis}@nii.ac.jp

ABSTRACT

We propose a neural network-based watermarking approach for defending against speaker identity manipulation attacks. Our method extracts a source speaker embedding from a carrier waveform and embeds it back into the waveform before transmission. After undergoing various channel transmissions and potential identity manipulation attacks, the receiver reconstructs the source speaker embedding from the extracted watermark and compares it with the embedding obtained from the received waveform to assess the likelihood of identity manipulation. Experimental results demonstrate the robustness of the proposed framework against multiple digital signal processing based transmissions and attacks. However, we observe that while neural codec algorithms have minimal impact on manipulating speaker identity, they significantly degrade watermark detection accuracy, leading to failures in detecting identity manipulation.

1 INTRODUCTION

Physiological and psychological characteristics of a person's speech can be represented as high-dimensional, deep representations using machine learning models (Bai & Zhang, 2021; Wang et al., 2024b). These representations, known as speaker embeddings, play a key role in many speech processing tasks such as voice authentication (Snyder et al., 2018), generation (Ju et al., 2024), and conversion (Lu et al., 2019; Park et al., 2023). The widespread use of such embeddings also raises security and privacy concerns regarding their potential misuse (Wang et al., 2024b; Panariello et al., 2024).

Today, deep-learning-based zero-shot voice generation and conversion systems can seamlessly replace the voice in any given speech with another speaker's voice, using recordings as short as a few seconds (Ju et al., 2024; Chen et al., 2025). These regenerated speech samples are so natural and realistic that they can even outperform genuine ones in tasks such as speech recognition measured by word error rate (Eskimez et al., 2024). Simply relying on human auditory perception for determining whether a speech sample is genuine or synthetic is no longer sufficient, or more critically, reliable. Instead, we now have to rely on specifically trained, also deep-learning-based techniques (Li et al., 2024; 2025; Zhou et al., 2024b) for secure and robust detection of generated and partially-manipulated speech.

Protection against such deepfake speech falls into two categories: passive and proactive. Passive models, often binary classifiers (Li et al., 2025; Zhang et al., 2022), are trained using paired genuine and fake speech data. Proactive models, typically neural network-based watermarking (Chen et al., 2023; Roman et al., 2024b), embed imperceptible, multi-bit messages into speech. Naturally, watermarking models can carry more information than binary classifiers. Their multi-bit capacity can also be leveraged in other applications, such as steganography, data-hiding, and self-embedding (Fridrich & Goljan, 1999), where the goal is to conceal a watermark message for both content protection and recovery (Li et al., 2023; Chang et al., 2023).

In this work, we aim to embed speaker embedding into speech to defend against speaker identity manipulation. We propose a framework in which any receiver of potentially manipulated speech can verify, using only the received speech itself, whether the presented speaker matches the original one. In summary, the main contributions of this paper are:

- We propose a novel framework for detecting speaker identity manipulation, consisting of a speaker encoder for extracting speaker embeddings and a watermarking model for injecting and detecting these embedding.
- To ensure the speaker encoder remains compatible with the capacity constraints of the watermarking model, we apply Matryoshka Representation Learning (MRL) (Wang et al., 2024c) to hierarchically structure the speaker embeddings, allowing for dimensionality reduction while preserving key speaker identity information.
- We evaluate the proposed framework under various digital signal processing (DSP) and neural network (NN)-based transmission distortions, as well as identity manipulation attacks.

2 Related works

Speech watermarking generally shares the same objective as in other modalities like text (Liu et al., 2024a) and images (Wan et al., 2022), where an embedder is designed to inject imperceptible, multi-bit messages (watermarks) into the target carrier, and a detector is designed to accurately recover the message, even after various forgery or removal attacks (Roman et al., 2024b; Ji et al., 2025; Liu et al., 2024c). To achieve robust detection performance against such attacks, watermarks are typically embedded in the speech frequency domain (Chen et al., 2023; Liu et al., 2024b), and more recently, in deep latent representations (Roman et al., 2024a; Ji et al., 2025; Zhou et al., 2024a).

Steganography (Li et al., 2023; Chang et al., 2023), a special use case of watermarking, is the practice of hiding a secret message within a carrier without raising suspicion. The hidden message can either be an image, or text message, or file. In the context of speech, the message can be a compressed version of the original signal injected into each speech frame and serves to restore and recover signals after distortion (Wang et al., 2024a; Quiñonez-Carbajal et al., 2024). Unlike prior work focused on speech restoration of target segments, our approach utilizes watermarking to preserve speaker identity, ensuring security and traceability under identity manipulation attacks.

Speaker embedding differs from temporal speech information like content, which may vary within the same utterance. It is an utterance-level, speaker-specific representation (Desplanques et al., 2020; Snyder et al., 2018). Utterances spoken by the same speaker typically produce embeddings with high similarity, while embeddings from different speakers generally show lower similarity.

Speaker embedding can be captured in multiple ways. Traditional methods involve classification training on large, multi-speaker, multi-utterance datasets (Chung et al., 2018; Desplanques et al., 2020). Other approaches focus on disentangling attributes from a single speech sample, treating the global characteristics as a timbre representation (Ju et al., 2024). Recent NN-based manipulations of speaker identity often involve replacing speaker related representations with those of another speaker while preserving other attributes like prosody and content for reconstruction (Park et al., 2023; Champion et al., 2022).

3 Methodology

In Sec. 3.1, we introduce the proposed manipulation detection pipeline. We then provide a detailed description of the speaker encoder in Sec. 3.2, followed by an explanation of the watermarking process in Sec. 3.3. Lastly, we outline the transmission processing and attack scenarios used in our experiments in Sec. 3.4 and Sec. 3.5.

3.1 Proposed pipeline

Figure 1 illustrates the proposed detection pipeline, where we consider a scenario involving four parties:

1. The **publisher** collects the speech content, embeds a watermark in the speech signal and shares the watermarked speech on a public platform. This process is shown in the upper section of Figure 1. The embedded watermark message is correlated with speaker identity, ensuring that it carries speaker-specific information.

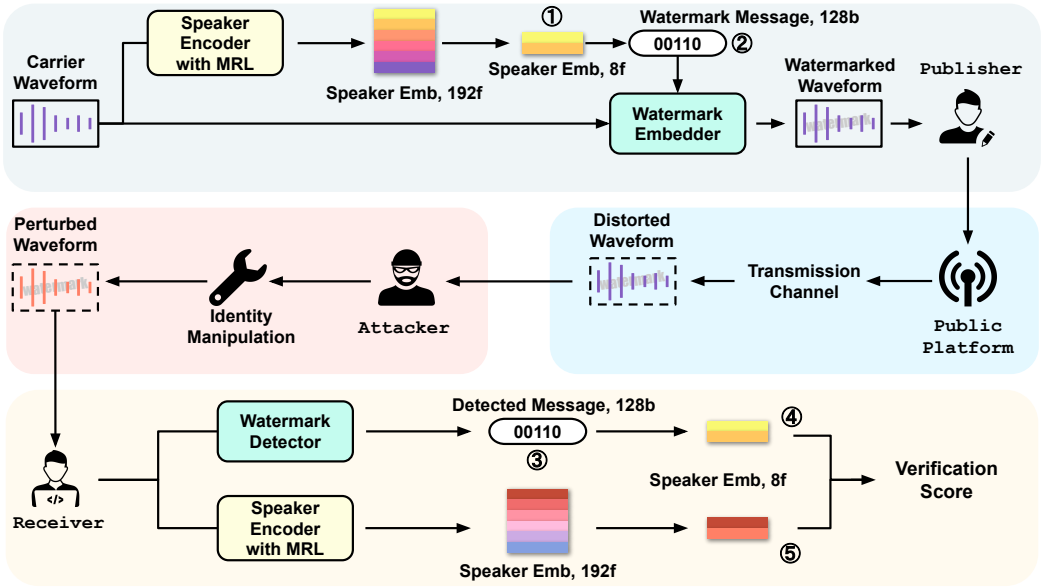

Figure 1: Proposed framework. Speaker embedding of original carrier waveform is first extracted using speaker encoder. From this embedding, only initial few dimensions (①) are selected and binarized to form watermark message (②). This message is then embedded into carrier waveform via watermark embedder. After transmission through public channel, waveform may undergo identity manipulation attacks. To verify authenticity, watermark detector decodes embedded message from potentially perturbed waveform, recovering detected message (③), which is then used to reconstruct recovered speaker embedding (④). Simultaneously, same speaker encoder extracts speaker embedding directly from distorted waveform (⑤). Similarity between embeddings ④ and ⑤ serves as score to determine whether source speaker's identity has been compromised. MRL indicates Matryoshka Representation Learning.

2. The **public platform** transmits the watermarked speech waveform through multiple channels to reach the receiver, as shown in the middle-right section of Figure 1. While the transmission process may degrade the speech quality or fidelity, it is assumed that the watermark message remains unaffected.
3. The **attacker** intercepts the speech and attempts to alter the speaker identity, replacing it with a different one. This manipulation is shown in the middle-left section of Figure 1. The attacker is unaware of the presence of a watermark in the speech, as well as the algorithm used for embedding it. Additionally, we assume that while the speaker identity is altered, the watermark message remains either entirely unaffected or experiences minimal distortion.
4. The **receiver** downloads the speech and verifies whether the speaker identity has been changed, as shown in the bottom section of Figure 1. By extracting and analyzing the embedded watermark, the receiver determines whether the original speaker identity has been compromised.

The watermark encoding and detection process involves the **publisher** and **receiver**. On the publisher side, the message to be embedded is obtained by feeding the carrier waveform into a speaker encoder to extract its speaker embedding. However, raw speaker embeddings are not suitable as watermarks since they contain floating-point numbers, while most watermarking methods require multi-bit binary messages. To address this and avoid the loss of speaker information when binarizing the entire embedding, we use only the initial part of the speaker embedding and binarize it to match the capacity of the watermark encoder model, which then injects the binarized message into the carrier waveform. Details are provided in Sec. 3.2 and Sec. 3.3.

On the receiver side, upon receiving the watermarked waveform, we first use a watermark detector to extract the watermark message and convert it back to floating-point numbers following the same

rules used during binarization. The same speaker encoder is then applied to extract the speaker embedding from the received waveform. Finally, we calculate the cosine similarity between the two embeddings as a detection score. A higher similarity indicates that the speaker in the received waveform matches the original speaker, while a lower similarity suggests a mismatch, indicating possible identity manipulation during transmission.

3.2 SPEAKER ENCODING AND BINARIZATION

We use ECAPA-TDNN[1] (Desplanques et al., 2020), a widely used model in the automatic speaker verification (ASV) community (Wang et al., 2024b), for the speaker encoder. The ECAPA-TDNN model mainly consists three SE-Res2Block (Hu et al., 2018; Gao et al., 2019) modules and is trained to associate input waveforms with their corresponding speakers. Since ASV training databases typically contain multiple speakers and multiple utterances per speaker, the ECAPA-TDNN model is optimized using a loss function that maximizes intra-speaker similarity while minimizing inter-speaker similarity (Wang et al., 2018). Our speaker encoder contains ECAPA-TDNN without its final classification head, and the output feature maps from all three SE-Res2Block modules are aggregated and mapped to a 192-dimensional vector, referred to as the speaker embedding.

However, storing a single floating-point number typically requires 32 bits (i.e., FP32 format), which already exceeds the capacity of most speech watermark models (10 or 16 bits, as in Roman et al. (2024b) and Chen et al. (2023)). While increasing the capacity is possible with a performance trade-off, storing the entire speaker embedding would require over 6,000 bits, making dimensionality reduction necessary.

To reduce the size of the message while maintaining its informativeness and allowing flexibility in choosing a message length (Wang et al., 2023; Fan et al., 2019), we apply Matryoshka Representation Learning (MRL) loss (Kusupati et al., 2022; Wang et al., 2024c) during training. Unlike the original ECAPA-TDNN training, where final classification head maps a 192-dimensional embedding to a vector corresponding to the total number of speakers in the dataset, MRL structures the embedding hierarchically. Instead of using a single classification head, MRL partitions the embedding into nested subsets of dimensions – in our case, $[8, 16, 32, 64, 128, 192]$ – with each subset mapped to its own classification head. The identification losses from all these classification heads are summed with equal weights to optimize the network parameters. This approach enforces a hierarchical structure within the speaker embedding, where later dimensions are built upon and complement earlier ones.

After training, we use only the leading portion of the speaker embedding to construct the watermark message. Specifically, we select the first 8 dimensions of the ECAPA-TDNN embedding, with each of these 8 values binarized using the FP16 format, where each floating-point number is converted into 16 bits. As a result, the final watermark message has a total length of $8 \times 16 = 128$ bits.

Our work does not focus on the selection of the speaker encoder model or the MRL loss. Instead, our goal is to present a structured embedding approach that balances dimensionality reduction with identity preservation, enabling effective watermarking. Our framework can also accommodate alternative speaker models and dimensionality reduction techniques, such as PCA, autoencoders, and quantization.

3.3 WATERMARK ENCODING AND DECODING

We use the Timbre watermarking model[2] (Liu et al., 2024b) for encoding and decoding binarized speaker embeddings due to its proven high capacity for message injection. Timbre consists of two main modules: a watermark embedder and a watermark detector.

The embedder embeds a watermark message in the frequency domain by applying the Short-Time Fourier Transform (STFT) to the carrier waveform, obtaining the carrier spectrogram $S \in \mathbb{R}^{F \times T \times 1}$, where F is the number of frequency bins, and T is the number of frames. The embedder encodes an N-bit message $w \in \mathbb{R}^{N \times 1 \times 1}$ into a watermark feature $f_w \in \mathbb{R}^{F \times 1 \times 1}$, which is then broadcasted

[1] https://github.com/TaoRuijie/ECAPA-TDNN

[2] https://github.com/TimbreWatermarking/TimbreWatermarking

along the time axis to form the watermark embedding $f_W \in \mathbb{R}^{F\times T\times 1}$. This repetition ensures the time-independence of the watermark message, making it robust to distortions in the time domain.

Additionally, the embedder encodes the carrier spectrogram S into a carrier feature $f_S \in \mathbb{R}^{F\times T\times D}$, where D is the channel dimension. It then concatenates the encoded features f_W and f_S along with the original carrier spectrogram S to form a deep representation $f_E \in \mathbb{R}^{F\times T\times (D+2)}$. Finally, the embedder maps f_E to produce the watermarked spectrogram $S_w \in \mathbb{R}^{F\times T\times 1}$. This new watermarked spectrogram, along with the original carrier phase, is used to synthesize the watermarked waveform using inverse STFT.

The Timbre detector receives the (potentially distorted) watermarked waveform and converts it into a spectrogram S'_w. It then extracts the watermark embedding f'_W and averages it along the time axis to form the estimated watermark feature f'_w. Finally, the detector decodes f'_w to reconstruct the watermark message w'. To enhance robustness against distortions such as speech editing (Roman et al., 2024b), voice cloning (Liu et al., 2024b), and real-world attacks (Zhou et al., 2024b), various attacks are added as data augmentation (referred to as a distortion layer) to the watermarked waveform before it reaches the detector.

The Timbre model is trained with multiple losses that regulate different aspects of its behavior (Liu et al., 2024b):

- a watermark reconstruction loss ensures accurate watermark recovery at the detector;
- a waveform reconstruction loss at the embedder ensures that the watermarked waveform maintains a high audio quality and that the watermark remains imperceptible;
- an adversarial loss at the embedder helps ensure the realism of the watermarked waveform.

We use the Timbre model as is, where the publisher operates the watermark embedder module, and the receiver operates the watermark detector module.

3.4 TRANSMISSIONS

In our proposed scenario, we simulate distortions in the watermarked waveform during its transmission. They include four DSP-based distortions: Gaussian noise, echo, waveform quantization and low-pass filtering, along with two NN-based codec methods: EnCodec (Défossez et al., 2023) and DAC (Kumar et al., 2023). Gaussian noise and echo simulate noisy channel conditions, while the remaining are selected for their ability to reduce transmission bandwidth.

Various settings are applied to each transmission:

- Echo is added with $\{0.1, 0.3, 0.5, 0.7\}$ seconds delay through a simulated impulse response.
- Gaussian noise is introduced at signal-to-noise ratios (SNRs) of $\{5, 10, 20, 40\}$ dB.
- Low-pass filtering uses cutoff frequencies of $\{1.6, 3.2, 4.8, 6.4\}$ kHz.
- Waveform quantization is performed at $\{8, 16, 32, 64\}$ bits.
- DAC[3] is applied with an 8-kbps bitrate.
- EnCodec[4] is applied with bitrates of $\{3, 6, 12, 24\}$ kbps.

3.5 IDENTITY MANIPULATION ATTACKS

We assume that the receiver has no prior knowledge of whether the received waveform has been perturbed by an attacker and always processes it in a default manner, i.e., a black-box scenario. We select distortions that significantly alter speaker embeddings as identity manipulation attacks. The effects of these distortions on speaker embedding similarity, before and after application, are listed in Table 1.

However, not all distortions have the same level of impact on speaker identity. For instance, an utterance before and after waveform clipping at 90% of its maximum amplitude retains a higher

[3] https://github.com/descriptinc/descript-audio-codec

[4] https://github.com/facebookresearch/encodec

Attack	Params	Similarity	Attack	Params	Similarity
Intra-Spk	-	0.62	Inter-Spk	-	0.05
Clipping	90%	0.76	Resample	22,050 Hz	**0.14**
Clipping	80%	**0.60**	Resample	8,000 Hz	**0.11**
Clipping	70%	**0.50**	Resample	4,000 Hz	**0.04**
Time stretch	0.7	0.70	Pitch shift	2	**0.53**
Time stretch	1.3	0.72	Pitch shift	4	**0.28**
Time stretch	0.9	0.76	Pitch shift	6	**0.18**
Time stretch	1.3	0.72	Pitch shift	8	**0.15**
kNN	-	**0.38**			

Table 1: Averaged speaker similarity scores under various distortions. Intra-Spk refers to similarity between two utterances spoken by same speaker. Inter-Spk refers to similarity between utterances spoken by different speakers. Similarity scores for other attacks are computed by comparing same utterance before and after applying distortion. Bold values indicate attack results where similarity is lower than Intra-Spk (0.62), meaning they significantly alter speaker identity.

similarity (0.76) than two different, unprocessed utterances from the same speaker (0.62). To ensure that the selected attacks cause a substantial degradation in speaker similarity, we define a threshold: the similarity after applying a distortion must be clearly lower than that of different utterances from the same speaker (i.e., Intra-Spk similarity).

On the basis of the results in Table 1, we select the following identity manipulation attacks:

- DSP-based attacks:
  - Clipping: Applied at an amplitude threshold of 70%.
  - Resampling: From 16 kHz to $\{4, 8, 22.05\}$ kHz.
  - Pitch Shift: Pitch increased by $\{2, 4, 6, 8\}$ semitones.
- NN-based Attack:
  - k-Nearest Neighbors Voice Conversion (kNN-VC) (Baas et al., 2023).

Resampling is performed only once on attacker's side, and receiver processes the resampled waveform directly without converting it back to 16 kHz. This process is similar to speed scaling used in Hua et al. (2016). kNN-VC performs voice conversion by replacing the frame-level deep representations of the source speech with their nearest neighbor in a reference speech pool. The converted waveform is then synthesized using a HiFi-GAN vocoder (Kong et al., 2020). We use the default settings for kNN-VC[5].

4 Experimental setup

This section outlines the datasets and metric used in this study.

4.1 Datasets

We utilized the VoxCeleb databases (Nagrani et al., 2017; Chung et al., 2018) for training and performance evaluation. These datasets contain publicly available, in-the-wild 16 kHz sampled voice recordings from more than 6,000 speakers. Such uncontrolled conditions are more reflective of real-world scenarios compared with clean, noise-free datasets. All reported results are based on the VoxCeleb1 test partition.

The ECAPA-TDNN model was trained with its default settings with MRL loss using the VoxCeleb2 development partition. Noise files from the MUSAN corpus (Snyder et al., 2015) and room impulse response (RIR) filters (Ko et al., 2017) were added during training for data augmentation. The best model was selected on the basis of the speaker verification performance on the VoxCeleb1 test partition.

[5] https://github.com/bshall/kNN-VC

The Timbre model was trained with its defaults settings using a random subset of 70,000 utterances from the VoxCeleb1 development partition for model learning. The transmissions described in Sec. 3.4, except for EnCodec, were used in the distortion layer to improve the detector performance. The best model was selected on the basis of the watermark recover accuracy on the VoxCeleb1 test partition.

For the voice conversion process involved in the kNN-VC attack, source utterances were converted to their nearest neighbors in the reference pool including a random selection of one utterance from all the speakers in the VoxCeleb2 test partition.

4.2 METRIC

We computed the cosine similarity between the speaker embeddings recovered from the watermark message and extracted from the perturbed waveform (marked as ④ and ⑤ in Figure 1) as the detection score. To evaluate performance, we used the equal error rate (EER) metric, which is widely used in the speaker verification community. It estimates the maximum probability of making an incorrect classification in the case of an optimal Bayes decision (Brummer, 2010):

- The **positive class** refers to cases where no attack has been applied to manipulate the speaker identity. That is, the recovered speaker embedding (④) and the extracted speaker embedding (⑤) are considered to originate from the same speaker.
- The **negative class** includes cases where both transmission distortions and identity manipulation attacks are present, causing ④ and ⑤ to be treated as embeddings from different speakers.

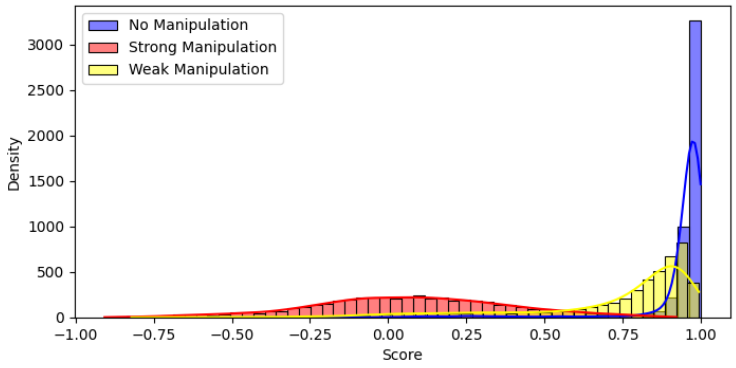

Figure 2: Distributions of scores used for EER estimations.

Figure 2 illustrates how the detection score distributions vary under different attack conditions. In the case of a mild identity manipulation attack (yellow), the speaker identity remains relatively similar to the original (blue). This leads to a higher EER, as distinguishing manipulated speech from the original becomes more difficult. In contrast, under an aggressive identity manipulation attack (red), the detection scores become more widely distributed, making it easier to separate manipulated speech from unaltered speech. As a result, the EER decreases.

5 RESULT AND ANALYSIS

Table 2 presents the EERs calculated on the basis of cosine similarity scores between the recovered embeddings from watermark messages and the speaker embeddings extracted from the watermarked and perturbed waveforms, denoted as ④ and ⑤ in Figure1, respectively. The table displays results across various transmission methods (rows) and attacks (columns). Transmissions are grouped as DSP-based and NN-based, while attacks are sorted by their effectiveness in manipulating speaker similarity. Additionally, cells are color-coded in gray-scale to reflect EER values: darker shades indicate higher EERs, while lighter shades represent lower EERs.

We first analyze the results presented in Table 2 attack-wisely in Sec. 5.1, followed by a transmission-wise analysis in Sec. 5.2. We discuss our findings in Sec. 5.3.

	Column ID		1	2	3	4	5	6	7	8	9
			Mild Identity Manipulation → Aggressive Identity Manipulation								
	Attack / Transmission	Params	Clipping	kNN	Pitch Shift				Resampling		
			70%	-	2	4	6	8	4 kHz	8 kHz	22.05 kHz
	None	-	1.11	0.02	0.02	0.06	0.06	0.04	0.00	0.04	0.04
DSP	Echo	0.1	3.14	0.27	0.23	0.37	0.29	0.37	0.00	0.14	0.29
DSP	Echo	0.3	2.65	0.21	0.16	0.27	0.27	0.25	0.18	0.18	0.23
DSP	Echo	0.5	2.48	0.10	0.06	0.18	0.23	0.14	0.00	0.12	0.14
DSP	Echo	0.7	2.59	0.18	0.10	0.31	0.29	0.31	0.04	0.14	0.27
DSP	Gaussian	5 dB	35.51	22.79	27.37	27.60	28.95	27.33	28.48	28.66	26.12
DSP	Gaussian	10 dB	22.26	14.61	15.57	16.84	16.78	16.00	16.41	16.66	15.43
DSP	Gaussian	20 dB	7.06	3.96	3.75	4.27	4.45	4.01	3.96	4.02	4.04
DSP	Gaussian	40 dB	1.27	0.06	0.06	0.12	0.08	0.14	0.00	0.00	0.14
DSP	Lowpass	1.6 kHz	45.92	34.51	41.01	40.13	42.53	33.63	41.36	42.06	43.76
DSP	Lowpass	3.2 kHz	35.49	25.97	24.83	27.51	30.20	32.38	28.15	27.86	31.53
DSP	Lowpass	4.8 kHz	8.41	5.38	4.94	6.22	5.97	5.79	5.17	5.46	5.58
DSP	Lowpass	6.4 kHz	0.82	0.14	0.08	0.27	0.16	0.14	0.02	0.10	0.12
DSP	Quantization	8-bit	34.41	26.53	29.71	30.76	31.37	32.58	31.08	31.58	30.14
DSP	Quantization	16-bit	20.48	15.51	15.55	16.25	16.02	15.98	16.04	16.70	16.04
DSP	Quantization	32-bit	8.27	5.79	5.58	6.52	6.36	5.79	5.68	6.01	6.05
DSP	Quantization	64-bit	4.04	2.61	2.24	2.48	2.67	2.34	2.07	2.48	2.28
NN	DAC	-	44.09	38.22	41.01	38.20	41.71	39.39	41.55	39.84	41.51
NN	EnCodec	3 kbps	54.37	57.35	51.66	55.17	58.64	65.31	59.77	64.28	49.20
NN	EnCodec	6 kbps	50.90	53.73	50.02	52.28	54.55	59.27	55.70	57.00	50.10
NN	EnCodec	12 kbps	49.96	53.34	49.55	53.10	54.92	58.25	54.29	55.21	54.66
NN	EnCodec	24 kbps	48.95	51.78	48.81	51.87	53.75	55.81	53.16	54.16	53.61

Table 2: EER results of identity manipulation detection using cosine similarity between ④ recovered embedding from watermark message and ⑤ speaker embedding from watermarked and perturbed waveform as illustrated in Figure 1. Darker cell color indicates higher EER result, hence worse detection performance. Cells with lighter background color are transmission-attack combinations that are easier for identity manipulation detection.

5.1 Attack analysis

When no transmission was applied, the detection EERs for almost all selected attacks were near zero, meaning that the proposed system can effectively detect identity manipulation in this case. Among the attacks, kNN (column 2), pitch shift with two steps (column 3), and resampling with 4 kHz (column 7) yield the lowest EER results in the table.

When transmissions were applied, EERs were noticeably higher than those without transmission. However, the attack-level EER results generally followed the trend of their impact on speaker embedding similarity, as shown in Table 1 – mild manipulation attacks resulted in higher EERs (darker cells) than aggressive manipulation attacks because they lead to a higher embedding similarity, making them more difficult to distinguish from the original. For attacks with multiple parameter settings (pitch shift in columns 3–6 and resampling in columns 7-9), more aggressive settings consistently yielded lower EERs.

5.2 Transmission analysis

We observed different trends for DSP-based and NN-based transmissions. The EER results under echo transmission were close to those without transmission, as echo has a very limited impact on speaker similarity – speaker embedding is a speaker-specific feature and echo does not introduce a new speaker. Among other DSP-based transmissions, those that obscure critical speaker information, such as low SNR Gaussian noise (i.e., larger noise), low cutoff frequency filtering (which cuts off speech formants), and low-bit quantization (i.e., large quantization noise), generally cause higher EERs.

Although these distortions are seen transmissions during Timbre training, allowing for near-perfect message detection and accurate reconstruction of the speaker embedding, the reconstructed embedding (a near-perfect ④) exhibits low similarity to the extracted speaker embedding (⑤) when the latter is perturbed by aggressive transmissions. While identity manipulation attacks further degrade

this similarity, the additional degradation is relatively minor, making it harder to differentiate manipulation attacks.

For NN-based transmissions, although the EER results for DAC were slightly lower than those for EnCodec, both approaches yielded EER values close to 50% across *all* attack types. This suggests that under codec transmissions, the system is unable to detect the presence of attacks.

5.3 Discussion

Although the clipping attack (column 1) also resulted in a near-50% EER under certain DSP-based transmissions, the poor performance of DAC and EnCodec appears to be attack-irrelevant. This is surprising, as both NN-based codecs are used solely for waveform reconstruction and have minimal impact on manipulating speaker identity.

This leads us to hypothesize that these methods disrupt the non-speaker-encoder component of the detection pipeline, specifically the watermark detecting part, which is assumed to be robust against transmission distortions (Sec. 3.1). Similar vulnerabilities have been reported in the literature, indicating that watermark models are susceptible to neural codec applications (Roman et al., 2024b; Juvela & Wang, 2025). Without additional augmentation during training, the performance of watermarking models can even degrade to random guessing under neural codec conditions (Roman et al., 2024b).

Since our EER results are always calculated under transmissions for both same-speaker and different-speaker cases (Sec. 4.2), even with a perfect speaker encoder that always extracts an identical speaker embedding (⑤ in Figure 1) for the same speaker, a near-randomly reconstructed speaker embedding (④ in Figure 1) will affect score distributions in both cases. This makes the distributions with and without an attack almost inseparable, as in either case, the extracted watermark and its reconstructed speaker embedding are almost random, leading to a lower but closely distributed embedding similarity score and, consequently, a higher EER. This also explains why EnCodec consistently produces high EER results, while DAC's results are slightly lower – EnCodec is the only unseen transmission during training, and it affects the reconstruction of ④ more than DAC.

6 Conclusion

We proposed a proactive defense approach against speaker identity manipulation. We use a watermark model for message hiding and extraction, where the message itself contains speaker information extracted by a speaker model. By doing so, receivers can extract both the watermark message and speaker embedding from the watermarked speech and verify whether the source speaker identity has been manipulated.

Through our analysis, we identified that manipulation attacks are relatively easier to detect if important speaker information is not distorted during transmission. We also observed that the proposed detection framework fails under the two neural codec transmission cases. Although these transmissions do not affect speaker information, their negative impact on the watermark detection part causes a near-random embedding reconstruction from a near-random watermark message, making the detection scores inseparable when comparing cases with and without an attack.

Improving the watermarking model's robustness against neural codecs is a clear future research direction. Additionally, due to the limited capacity of existing watermarking models, in the current work, we must specifically tune the speaker model to obtain a compact speaker embedding. Therefore, high-capacity watermarking that can cover full speaker embedding information is desirable. Finally, similar to injecting speaker information to defend against identity manipulation attacks, we can also embed content information to defend against content editing attacks.

Acknowledgments

This study is partially supported by JST AIP Acceleration Research (JPMJCR24U3), JST CREST Grants (JPMJCR20D3), JST PRESTO (JPMJPR23P9) and by MEXT KAKENHI Grants (24H00732)

REFERENCES

Matthew Baas, Benjamin van Niekerk, and Herman Kamper. Voice conversion with just nearest neighbors. In *INTERSPEECH 2023*, pp. 2053–2057, 2023.

Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140:65–99, 2021.

Niko Brummer. *Measuring, refining and calibrating speaker and language information extracted from speech*. PhD thesis, Stellenbosch: University of Stellenbosch, 2010.

Pierre Champion, Anthony Larcher, and Denis Jouvet. Are disentangled representations all you need to build speaker anonymization systems? In *INTERSPEECH 2022*, pp. 2793–2797, 2022.

Ching-Chun Chang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Cyber vaccine for deepfake immunity. *IEEE Access*, 11:105027–105039, 2023.

Guangyu Chen, Yu Wu, Shujie Liu, Tao Liu, Xiaoyong Du, and Furu Wei. WavMark: Watermarking for audio generation. *arXiv preprint arXiv:2308.12770*, 2023.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural Codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1–15, 2025.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH 2018*, pp. 1086–1090, 2018.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. pp. 3830–3834, 2020.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. E2 TTS: Embarrassingly easy fully non-autoregressive zero-shot TTS. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 682–689, 2024.

Lei Fan, Qing-Yuan Jiang, Ya-Qi Yu, and Wu-Jun Li. Deep hashing for speaker identification and retrieval. In *INTERSPEECH 2019*, pp. 2908–2912, 2019.

Jiri Fridrich and Miroslav Goljan. Images with self-correcting capabilities. In *Proceedings 1999 International conference on image processing*, volume 3, pp. 792–796, 1999.

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2Net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Guang Hua, Jiwu Huang, Yun Q. Shi, Jonathan Goh, and Vrizlynn L.L. Thing. Twenty years of digital audio watermarking—a comprehensive review. *Signal Processing*, 128:222–242, 2016.

Shengpeng Ji, Ziyue Jiang, Jialong Zuo, Minghui Fang, Yifu Chen, Tao Jin, and Zhou Zhao. Speech watermarking with discrete intermediate representations. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiang-Yang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

Lauri Juvela and Xin Wang. Audio Codec augmentation for robust collaborative watermarking of speech synthesis. In *2025 IEEE international conference on acoustics, speech and signal processing (ICASSP 2025)*, 2025.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP 2017)*, pp. 5220–5224, 2017.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Menglu Li, Yasaman Ahmadiadli, and Xiao-Ping Zhang. A survey on speech deepfake detection. *ACM Computing Surveys*, 2025.

Songbin Li, Jingang Wang, Peng Liu, and Ke Shi. SANet: A compressed speech encoder and steganography algorithm independent steganalysis deep neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:680–690, 2023.

Ze Li, Yuke Lin, Tian Yao, Hongbin Suo, Pengyuan Zhang, Yanzhen Ren, Zexin Cai, Hiromitsu Nishizaki, and Ming Li. The database and benchmark for the source speaker tracing challenge 2024. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1254–1261, 2024.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024a.

Chang Liu, Jie Zhang, Tianwei Zhang, Xi Yang, Weiming Zhang, and Nenghai Yu. Detecting voice cloning attacks via timbre watermarking. In *Network and Distributed System Security Symposium*, 2024b.

Hongbin Liu, Moyang Guo, Zhengyuan Jiang, Lun Wang, and Neil Zhenqiang Gong. AudioMarkBench: Benchmarking robustness of audio watermarking. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024c.

Hui Lu, Zhiyong Wu, Dongyang Dai, Runnan Li, Shiyin Kang, Jia Jia, and Helen Meng. One-shot voice conversion with global speaker embeddings. In *INTERSPEECH 2019*, pp. 669–673, 2019.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *INTERSPEECH 2017*, pp. 2616–2620, 2017.

Michele Panariello, Natalia Tomashenko, Xin Wang, Xiaoxiao Miao, Pierre Champion, Hubert Nourtel, Massimiliano Todisco, Nicholas Evans, Emmanuel Vincent, and Junichi Yamagishi. The VoicePrivacy 2022 Challenge: Progress and perspectives in voice anonymisation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Hyun Joon Park, Seok Woo Yang, Jin Sob Kim, Wooseok Shin, and Sung Won Han. TriAAN-VC: Triple adaptive attention normalization for any-to-any voice conversion. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, pp. 1–5. IEEE, 2023.

Maria T Quiñonez-Carbajal, Rogelio Reyes-Reyes, Volodymyr Ponomaryov, Clara Cruz-Ramos, and Beatriz P Garcia-Salgado. Speech signal authentication and self-recovery based on DTWT and ADPCM. *Multimedia Tools and Applications*, pp. 1–25, 2024.

Robin San Roman, Pierre Fernandez, Antoine Deleforge, Yossi Adi, and Romain Serizel. Latent watermarking of audio generative models. *arXiv preprint arXiv:2409.02915*, 2024a.

Robin San Roman, Pierre Fernandez, Hady Elsahar, Alexandre Défossez, Teddy Furon, and Tuan Tran. Proactive detection of voice cloning with localized watermarking. In *International Conference on Machine Learning*, volume 235, 2024b.

David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pp. 5329–5333, 2018.

Wenbo Wan, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488:226–247, 2022.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

Jiaying Wang, Xianglong Wang, Namin Wang, Lantian Li, and Dong Wang. Ordered and binary speaker embedding. In *INTERSPEECH 2023*, pp. 4683–4687, 2023.

Shengbei Wang, Weitao Yuan, Zhen Zhang, and Lin Wang. Speech watermarking based tamper detection and recovery scheme with high tolerable tamper rate. *Multimedia Tools and Applications*, 83(3):6711–6729, 2024a.

Shuai Wang, Zhengyang Chen, Kong Aik Lee, Yanmin Qian, and Haizhou Li. Overview of speaker modeling and its applications: From the lens of deep speaker representation learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4971–4998, 2024b.

Shuai Wang, Pengcheng Zhu, and Haizhou Li. M-Vec: Matryoshka speaker embeddings with flexible dimensions. *arXiv preprint arXiv:2409.15782*, 2024c.

Lin Zhang, Xin Wang, Erica Cooper, Nicholas Evans, and Junichi Yamagishi. The PartialSpoof database and countermeasures for the detection of short fake speech segments embedded in an utterance. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:813–825, 2022.

Junzuo Zhou, Jiangyan Yi, Yong Ren, Jianhua Tao, Tao Wang, and Chu Yuan Zhang. WMCodec: End-to-end neural speech codec with deep watermarking for authenticity verification. *arXiv preprint arXiv:2409.12121*, 2024a.

Junzuo Zhou, Jiangyan Yi, Tao Wang, Jianhua Tao, Ye Bai, Chu Yuan Zhang, Yong Ren, and Zhengqi Wen. TraceableSpeech: Towards proactively traceable text-to-speech with watermarking. In *INTERSPEECH 2024*, pp. 2250–2254, 2024b.