Transferability of Adversarial Examples across Speaker Embedding Models for Voice Privacy Protection

Kotaro Nakamura*, Takuya Takahashi* and Toru Nakashika*
* The University of Electro-Communications, Japan
E-mail: {k.nakamura, takahashi, nakashika}@uec.ac.jp Tel/Fax: +81-042-443-5602

***Abstract*—The widespread use of speaker identification technology has raised concerns regarding personal information contained in voice data, particularly voice privacy. While the application of adversarial attacks is a potential protection measure for this issue, the extent to which their attack effectiveness is maintained across speaker embedding extractors with different architectures (a property known as transferability) is not yet well understood. In this study, we experimentally evaluated the transferability of adversarial audio generated using the I-FGSM method. The evaluation targeted three types of speaker embedding extractors with different architectures: X-vector, Titanet, MelStyleEncoder and ECAPA-TDNN. We evaluated attack effectiveness using identification accuracy and the Equal Error Rate, and assessed audio quality using PESQ, which predicts subjective listening quality, and SI-SDR, which measures signal fidelity. The experimental results showed that in black-box attacks, attacks from models like X-vector to ECAPA-TDNN exhibited high transferability, while an asymmetrical relationship was observed where the reverse direction had low transferability. Furthermore, a clear trade-off between attack strength and audio quality was confirmed. These results indicate that adversarial audio can affect not only a single model but also other models, suggesting its effectiveness as a technology for voice privacy protection.**

I. Introduction

In recent years, the performance of speaker recognition technology, used to identify a person from their voice, has improved significantly, driven by developments in deep neural networks (DNNs) [1] . Modern speaker identification systems mainly use speaker embedding techniques, such as the X-vector [2], to convert a voice recording of any length into a fixed-size vector that captures a speaker's unique characteristics. However, this technological progress has also increased the risk to voice privacy. The increased accessibility and improved performance of DNN-based automatic speaker verification (ASV) systems leads to a situation that malicious actors can more easily exploit them, such as by using voice recordings for unauthorized identification or by inputting them into speaker verification systems [3] [4].

As a measure to address this issue, this research focuses on a technology that applies the principles of adversarial example [5] [6] [7]. This involves intentionally adding small, human-imperceptible perturbations to the original audio to cause an AI model to produce an incorrect result. Our aim is to use this technology to protect voice privacy by causing a speaker identification model to misidentify the original speaker, thus preventing their identity from being compromised.

For this privacy protection technology to be effective in the real world, transferability (the ability of an adversarial example generated for one model to remain effective against other, unknown models with different architectures) is a critical property. Previous studies have focused on white box attacks [8] [9], where the attacker has full knowledge of the model, and black box attacks [10] [11], which target a single model architecture without knowing internal details. However, transferability, which is essential for understanding the true threat and potential of adversarial audio, has not yet been fully studied.

This paper aims to fill this knowledge gap by experimentally investigating the transferability of adversarial audio across different types of speaker encoders. Through this investigation, we seek to clarify the feasibility of protecting voice privacy using adversarial examples from the perspective of their attack transferability. The findings of this study will not only contribute to a realistic risk assessment of ASV systems and the development of defense methods, but also suggest the potential for applying the fundamental effect of adversarial samples to various other voice-related tasks.

II. Adversarial example

Recent research into protecting speech privacy has focused on using adversarial samples. This involves adding small amounts of noise to the input speech to mislead the model. This enables the speaker's features to be hidden while maintaining the naturalness of the speech.

The fast gradient sign method (FGSM) [12] was proposed by Goodfellow et al. as a simple and efficient method for generating adversarial examples. FGSM is intentionally perturbs input data using gradient information from the model's loss function, causing the model to output incorrect predictions. Because computing is fast and simple, FGSM used widely in research on adversarial attacks and in the evaluation of model robustness.

The procedure for generating an adversarial example using FGSM is as follows. First, we define the loss function$L(\boldsymbol{x}, y, \theta)$ for a deep learning model, where $\boldsymbol{x}$ is the input data, y is the corresponding true label for the input, and

979-8-3315-7206-8/25/$31.00

θ is the model parameters. The gradient of the loss function with respect to this input $\boldsymbol{x}$ is calculated as follows:

$$\boldsymbol{g} = \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y, \theta) \tag{1}$$

Using the gradient of this loss function, an adversarial example $\boldsymbol{x}'$can be generated by adding a perturbation ϵ to the original input $\boldsymbol{x}$.

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \cdot \mathrm{sign}(\boldsymbol{g}) \tag{2}$$

where $\mathrm{sign(g)}$ represents the sign of the gradient of the loss function, taking a value of +1 or -1, and ϵ is a hyperparameter that controls the magnitude of the perturbation and typically takes a small positive value.

The iterative fast gradient sign method (I-FGSM) [13] was proposed as an improved version of FGSM. The specific improvement is that while FGSM performs its update in a single, large step, I-FGSM repeats the update multiple steps with a smaller step size. This iterative process allows for a more effective attack. I-FGSM recalculates the gradient at each small step, allowing it to more precisely follow the contours of the loss landscape. This finer-grained optimization enables the discovery of a more potent adversarial example that would be missed by a single-step method. I-FGSM is characterized by adding small perturbations to the input data using the gradient of the loss function at each update step, generating the adversarial example within a constrained range. The attack success rate is higher than that of FGSM, and its success rate is improved for many models.

The iterative step is represented by the following equation, which generates the perturbation incrementally over multiple iterations:

$$\boldsymbol{x}'_{t+1} = \boldsymbol{x}'_t + \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}'_t} L(\boldsymbol{x}'_t, y, \theta)) \tag{3}$$

where t is the step number, the initial value $\boldsymbol{x}'_0 = \boldsymbol{x}$, α is the magnitude of the perturbation at each step (step size), and $\mathrm{sign}(\nabla_{\boldsymbol{x}'_t} L(\boldsymbol{x}'_t, y, \theta))$represents the sign of the gradient of the loss function. After each update step, the generated adversarial example is constrained to maintain $-\epsilon \leq \boldsymbol{x}'_{t+1} \leq \epsilon$.

$$\boldsymbol{x}'_{t+1} = \mathrm{Clip}_{\epsilon}(\boldsymbol{x}'_{t+1}) \tag{4}$$

Clip is defined as the process of bringing an input number within a predetermined range of values, with a minimum and maximum value that are established in advance. The process terminates when the number of iterations reaches a maximum value T, or when the perturbation has sufficiently achieved the goal of maximizing the loss function.

Compared to FGSM, I-FGSM can generate more effective adversarial examples because it generates perturbations over multiple steps. Furthermore, by applying constraints to the adversarial example after each update, it is possible to maintain similarity between the original input and the adversarial example. However, due to the multi-step updates, the computation time increases, and this problem becomes particularly noticeable when applying it to high-resolution input data or large-scale datasets.

III. Speaker Encoders

The performance of modern speaker verification systems, which are based on deep learning, depends on their ability to extract speaker embeddings. In this study, we evaluate four representative speaker encoder models: X-vector, ECAPA-TDNN, MelStyleEncoder and TitaNet. These models were specifically chosen because their fundamentally different architectural designs provide a robust and diverse testbed for investigating the transferability of adversarial attacks.

A. X-vector

The X-vector [2] architecture is based on a Time Delay Neural Network (TDNN), a network design effective at capturing features with wide temporal context from sequential data like audio. Its structure consists of several TDNN layers that extract these frame-level features, followed by a pooling layer that aggregates the entire sequence to handle inputs of variable length.

A defining characteristic of the X-vector is its use of statistical pooling. This method summarizes the frame-level features by calculating not only their mean, which captures the central acoustic characteristics of the speaker, but also their standard deviation, which represents the variation and dynamics of those characteristics. These two statistics are then concatenated to form a single utterance-level vector that provides a richer and more robust representation of the speaker's identity for the entire recording.

B. ECAPA-TDNN

ECAPA-TDNN [14] is a high-performance speaker embedding model that significantly enhances the traditional TDNN architecture. Its overall structure is composed of three main stages: (1) a feature encoder that extracts deep, multi-scale features from the input audio, (2) an attention-based pooling layer that aggregates these features, and (3) an output layer that produces the final speaker embedding.

The core of the feature encoder is a stack of 'SE-Res2Blocks.' This specialized block is designed to capture patterns of various time lengths simultaneously. It achieves this by splitting the input channels into multiple segments and processing each with different temporal contexts, effectively capturing multi-scale features from the audio signal. Furthermore, each block integrates Squeeze-and-Excitation (SE) channel attention, which learns to emphasize the most important feature channels.

Finally, the attentional statistics pooling layer aggregates the frame-level features from the encoder. Unlike standard pooling that treats all frames equally, this layer intelligently assigns a learned weight to each frame, allowing the model to focus on the most speaker-discriminative segments of the utterance. This process results in a more robust and accurate utterance-level representation.

C. Titanet

TitaNet[15] is an architecture based on a Convolutional Neural Network (CNN), designed for both high performance

and computational efficiency. Its structure is primarily built from blocks of 1D depth-wise separable convolutions, which are a type of convolution that reduces computational complexity. Each block also includes an SE block and a residual connection. A defining feature of TitaNet is its use of a self-attention mechanism in its later layers. This mechanism allows the model to effectively learn long-range dependencies across the entire utterance, which complements the local patterns captured by the convolutional layers.

D. MelStyleEncoder

The MelStyleEncoder [16] is a key component of the StyleSpeech model [16] , designed to extract style characteristics from a reference audio and encode them into a fixed-length style embedding. It begins by taking the mel-spectrogram of a reference audio clip as input. This spectrogram is then passed through a stack of 2D convolutional layers followed by a Gated Recurrent Unit (GRU) layer. This combination of convolutional and recurrent layers is effective at capturing the complex, time-varying stylistic features from the audio signal. Finally, the features processed through these layers are aggregated and condensed into a single fixed-length style vector.

The most significant feature of the StyleSpeech is the ability to learn speaker styles in an unsupervised manner. The extracted style vector is utilized within the TTS generator via a mechanism called Style-Adaptive Layer Normalization (SALN). SALN's function is to adjust the properties of the text-based features according to the style vector, effectively infusing the synthesized speech with the style of the reference audio. A key advantage of this architecture is its capacity for high-quality, fast adaptation, enabling the model to perform voice cloning from just a single, short reference audio clip of 1-3 seconds. An extension of the model, called Meta-StyleSpeech, introduces Style Prototypes to further enhance performance. The MelStyleEncoder then learns to represent a new, unseen speaker's style as a combination of these prototypes, which improves its ability to adapt to new speakers.

IV. Methods

This study aims to evaluate the attack transferability of adversarial audio among four speaker encoders with different architectures and to clarify its effectiveness as a voice privacy protection technology. As for the method, we generate adversarial audio for each model using the I-FGSM algorithm and a loss function based on negative cosine similarity. We then analyze the strength of transferability, its asymmetry, and the trade-off with quality by cross-evaluating the attack performance (Accuracy, EER) and audio quality (PESQ, SI-SDR) of the generated audio among the models.

A. Datasets

In this study, experiments were conducted using the train-clean-100 subset of the LibriTTS [17] corpus for training each speaker identification model. The number of utterances of the 247 speakers in train-clean-100 was 200 or less, and 146 speakers were considered as target speakers. Furthermore, the number of utterances for each speaker was aligned to 200, for a total of 29,200 utterances used. In addition, 10 utterances were randomly selected from each speaker as test data for the model, for a total of 1,460 utterances. The remaining 27,740 utterances were used for training.

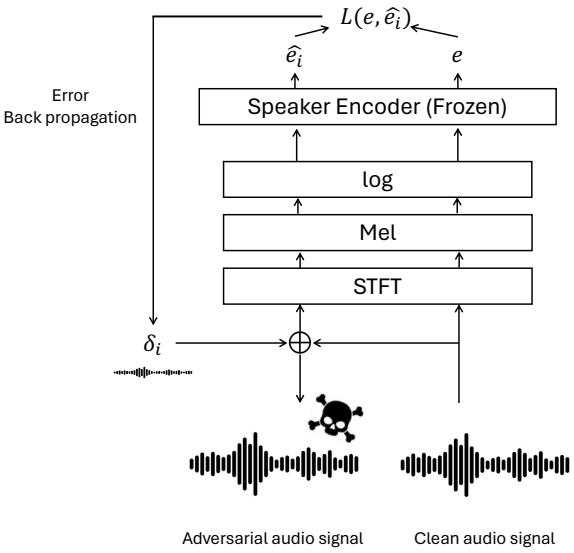

Fig. 1: Flow of adversarial voice generation

For the generation of adversarial audio, 10 speakers were randomly selected from the test data, and noise was added to 10 utterances from each of these speakers (100 utterances in total) to investigate the impact on the speaker identification models.

B. Adversarial audio generation

For the generation of adversarial audio, we employ the I-FGSM (iterative fast gradient sign method). To create an adversarial example, adversarial perturbation is added to the original audio signal.

The optimization of this noise is guided by a loss function. Specifically, we use cosine similarity as the loss function. Cosine similarity is a metric that measures the similarity between two vectors; in this case, it measures the similarity between the speaker embedding vectors of the original and the perturbed audio. The process aims to guide the speaker embedding vector of the perturbed audio to have different characteristics from that of the original audio. The loss function is defined as follows:

$$L = -\frac{\boldsymbol{e}^T \hat{\boldsymbol{e}}_i}{\|\boldsymbol{e}\| \|\hat{\boldsymbol{e}}_i\|} \tag{5}$$

The generation process for the adversarial audio is illustrated in Figure 1. In this process, the terms $\boldsymbol{e}$ and $\hat{\boldsymbol{e}}_i$ are the speaker embedding vectors for the original and perturbed audio, respectively. L denotes the loss function, which is the negative cosine similarity, and δ_i represents the updated noise at each iteration.

TABLE I: Accuracy and EER of White and black box

Models generated adversarial examples	Iteration	X-vector		ECAPA-TDNN		MelStyleEncoder		Titanet	
		Accuracy(↓)	EER(↑)	Accuracy(↓)	EER(↑)	Accuracy(↓)	EER(↑)	Accuracy(↓)	EER(↑)
	0	0.98	0.127	0.99	0.064	0.76	0.241	0.99	0.029
X-vector	20	0.08	0.345	0.89	0.123	0.63	0.411	0.84	0.054
	40	0.04	0.429	0.64	0.155	0.61	0.407	0.74	0.078
	60	0.08	0.437	0.55	0.182	0.60	0.419	0.61	0.102
	80	0.06	0.398	0.45	0.194	0.57	0.432	0.52	0.101
	100	0.05	0.389	0.32	0.188	0.53	0.412	0.51	0.106
ECAPA-TDNN	20	0.85	0.206	0.16	0.320	0.67	0.410	0.97	0.037
	40	0.80	0.227	0.14	0.273	0.62	0.403	0.98	0.046
	60	0.78	0.241	0.14	0.269	0.58	0.404	0.97	0.045
	80	0.76	0.244	0.14	0.258	0.56	0.406	0.96	0.052
	100	0.75	0.246	0.14	0.252	0.57	0.415	0.97	0.047
MelStyleEncoder	20	0.93	0.143	0.99	0.078	0.30	0.445	0.99	0.026
	40	0.89	0.159	0.99	0.069	0.18	0.468	0.98	0.025
	60	0.88	0.164	0.99	0.080	0.14	0.466	0.97	0.024
	80	0.88	0.170	0.98	0.083	0.12	0.459	0.97	0.023
	100	0.86	0.175	0.96	0.086	0.11	0.451	0.97	0.026

To analyze the effect of the optimization process, the number of iterations for generating the adversarial noise was tested at values ranging from 10 to 100 with a step size of 10. For the I-FGSM method, the parameters were configured as follows: the maximum perturbation magnitude, ϵ, was set to 0.001, and the step size, α, was set to $\epsilon/10$=0.0001.

C. Evaluation index

The evaluation of attacks on the speaker identification models by adversarial audio is conducted using Accuracy and EER (Equal Error Rate).

Accuracy in this study is the percentage of speakers correctly predicted by the speaker identification model. A higher value indicates that the attack was successful. When the number of samples input to the model is N_{input} and the number of correctly authenticated samples is N_{success}, accuracy is defined by the following equation:

$$\text{Accuracy} = \frac{N_{\text{success}}}{N_{\text{input}}} \tag{6}$$

EER is a metric used to measure the misidentification rate of the speaker identification model, where a higher value indicates that the model's performance has degraded. EER refers to the value where the False Rejection Rate (FRR) and the False Acceptance Rate (FAR) are equal. The False Rejection Rate is the probability that a genuine user is incorrectly rejected, while the False Acceptance Rate is the probability that an imposter is authenticated as the genuine user.

As an evaluation metric, we use SI-SDR, which was proposed by Le Roux et al [18]. Traditional SDR has an issue where the score decreases merely due to a change in the overall volume of the estimated signal. To solve this problem, SI-SDR first optimally aligns the scale of the estimated signal with the ground-truth signal. It then calculates the power ratio between the correctly separated target signal component and the error component, which includes noise and distortion. This allows for a more robust evaluation of the separation or enhancement quality, without being affected by simple differences in scale. A higher value of the metric indicates better performance. The SI-SDR is formulated as following

$$\text{SI-SDR} = 10\log_{10}\left(\frac{\|\boldsymbol{x}_{target}\|^2}{\|\boldsymbol{x}_{noise}\|^2}\right) \tag{7}$$

where, $\|\boldsymbol{x}_{target}\|^2$ is a power of the target signal, and $\|\boldsymbol{x}_{noise}\|^2$ is a power of the error signal.

For the objective evaluation of speech quality, we also use the Perceptual Evaluation of Speech Quality (PESQ). PESQ is an algorithm standardized as ITU-T Recommendation P.862. The output is a score, typically on a scale from -0.5 to 4.5, where a higher score indicates a better perceived quality that is closer to the original signal.

V. Results

Table I shows the results of adversarial attack effectiveness, evaluated across the four speaker identification models: X-vector, ECAPA-TDNN, MelStyleEncoder and TitaNet. The table summarizes the identification accuracy and the Equal Error Rate (EER) that result when adversarial audio, generated from a source model, is input to a target model.

A. White box attack

The diagonal elements in Table I show the results of the white-box attacks, where the source and target models are identical. For the X-vector model, the accuracy was already 8% and the EER was 34.5% at just 20 iterations. For the ECAPA-TDNN model, the accuracy was consistently 14%, and the EER was approximately 24%. In MelStyleEncoder, accuracy is gradually decreasing, and it can be found that it has high attack performance even at low iterations. Comparing these three results, it was observed that the attack effectiveness in the white-box setting was higher for the X-vector model.

B. Black box attack

The off-diagonal elements are the results of black-box attacks, showing whether an attack generated on one model is also effective against other, different models (transferability). Looking at these results, it is clear that the adversarial

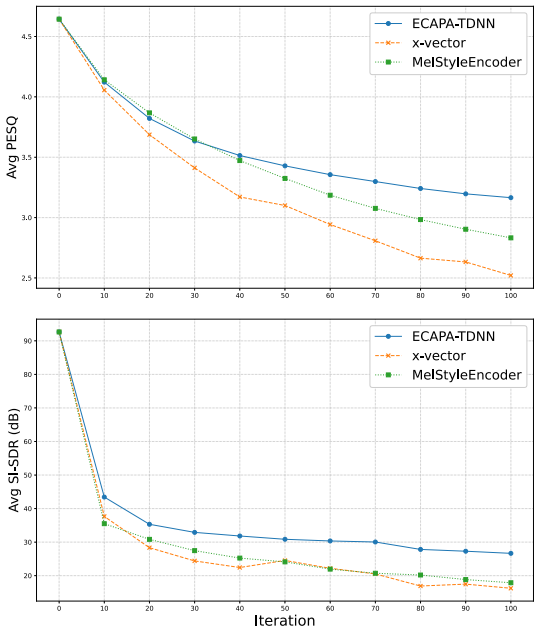

Fig. 2: PESQ and SI-SDR in each iteration

audio generated by X-vector has high attack effectiveness against TitaNet, MelStyleEncoder and ECAPA-TDNN, and that the adversarial audio generated by ECAPA-TDNN and MelStyleEncoder have an accuracy of over 90% when attacking TitaNet, showing that transferability is not observed.

C. Audio quality evaluation

Figure 2 shows the relationship between the number of iterations used to generate adversarial audio and the objective quality of the resulting audio for the three models: ECAPA-TDNN, X-vector, and MelStyleEncoder. The top graph plots the results for PESQ , and the bottom graph plots the results for SI-SDR.

From the graphs, a common trend is observed for all three models where both the PESQ scores and SI-SDR values decrease as the number of iterations increases. However, clear differences in performance among the models were also observed. For both metrics, ECAPA-TDNN consistently maintains the highest scores, indicating the best audio quality preservation under attack.

The results for MelStyleEncoder show a mixed performance relative to X-vector. In the PESQ evaluation (top graph), MelStyleEncoder consistently outperforms X-vector, positioning its quality between ECAPA-TDNN and X-vector. In contrast, for the SI-SDR evaluation (bottom graph), the performance of MelStyleEncoder is comparable to, and at several iteration points slightly lower than, that of X-vector.

Although the audio quality degrades for all models as the strength of the adversarial attack increases, the results objectively show that the adversarial audio generated with ECAPA-TDNN consistently maintains the highest quality. Meanwhile, the quality of audio from MelStyleEncoder is higher than that of X-vector according to the PESQ metric, but is similarly low when measured by the SI-SDR metric.

VI. Discussion

We discuss why the X-vector model may show better attack transferability than its successors, ECAPA-TDNN and Titanet, and why MelStyleEncoder shows poor transferability.

The embedding dimensionality appears to be a key factor. The high-dimensional, 512-dimensional space of the X-vector model may hold speaker characteristics in a more general and distributed manner compared to the 192-dimensional spaces of ECAPA-TDNN and Titanet, and the even lower 128-dimensional space of MelStyleEncoder. This suggests that in higher dimensions, the essential features of a speaker's identity are represented in a more fundamental form, rather than being highly specialized in a few dimensions. Consequently, an adversarial attack targeting this diffuse and fundamental representation is less likely to overfit to a specific architecture. This hypothesis aligns with our results, where attacks generated from the higher-dimensional X-vector showed strong transferability, while those from the lowest-dimensional MelStyleEncoder were the least transferable.

Additionally, architectural differences play a crucial role. The simple TDNN structure of the X-vector likely captures fundamental time-series patterns. An attack targeting these patterns remains effective against more complex models like ECAPA-TDNN and Titanet, as they must also process these same basic features. Conversely, an attack on ECAPA-TDNN or Titanet may target their advanced, specialized components (e.g., attention mechanisms), which do not exist in the simpler X-vector. This suggests an asymmetric transferability, where attacks on fundamental features are more portable.

The case of MelStyleEncoder further supports this architectural hypothesis. Unlike the other models trained for speaker identification, MelStyleEncoder is designed to capture broader style characteristics. It is plausible that perturbations optimized to disrupt these abstract style features do not effectively disrupt the identity features that the other models rely on. This specialization of its learning objective likely explains why attacks generated from MelStyleEncoder exhibit very poor transferability to the other models, as shown in the results.

VII. Conclusions

The purpose of this study was to evaluate the attack transferability of adversarial audio between speaker embedding extractors with different architectures and to verify its effectiveness as a means of voice privacy protection. Based on the mutual evaluation of adversarial audio generated using I-FGSM across four models.

First, in the white-box attacks, a significant difference in the vulnerability of the models was observed. The X-vector and the MelStyleEncoder models were highly vulnerable to attacks on itself, whereas the ECAPA-TDNN model demonstrated relatively high robustness. Second, in the black-box attacks, it was shown that a certain degree of transferability exists. It was also confirmed that there is an asymmetrical

relationship in the transferability of the attacks, meaning an attack successful in one direction does not guarantee success in the reverse direction. Third, the audio quality evaluation revealed a clear trade-off between the number of iterations and quality degradation. Furthermore, it was suggested that the ECAPA-TDNN model could generate adversarial audio with equivalent or greater attack effectiveness while maintaining higher audio quality than the X-vector model.

These results demonstrate that adversarial audio is an effective means of disrupting speaker information and that it can affect not only a specific model but also models with different architectures. This suggests that adversarial audio is a promising technology for voice privacy protection.

As for future work, possibilities include evaluating transferability against a more diverse range of model architectures, developing attack methods that further enhance transferability, and exploring techniques to maximize attack effectiveness while minimizing audio quality degradation.

ACKNOWLEDGMENT

This research was partly funded by JSPS Grants-in-Aid for Scientific Research 24H00715.

REFERENCES

[1] C. R. K. *et al*, "Ai and ml based google assistant for an organization using google cloud platform and dialogflow," *IJRTE*, vol. 8(5), pp. 2722–2727, 2020.

[2] e. a. D. Snyder, "X-vectors: Robust dnn embeddings for speaker recognition," *International Conference on Acoustic, Speech and Signal Processing(ICASSP)*, 2018.

[3] C.-Y. Y. *et al*, "Rw-voiceshield: Raw waveform-based adversarial attack on one-shot voice conversion," *Interspeech*, 2024.

[4] e. a. Rui Wang, "Asynchronous voice anonymization using adversarial perturbation on speaker embedding," *Interspeech*, 2024.

[5] e. a. Christian Szegedy, "Intriguing properties of neural networks," *in International Conference on Learning Representations (ICLR)*, 2014.

[6] B. S. Kurakin A Goodfellow I, "Adversarial examples in the physical world," *in International Conference on Learning Representations(ICLR)*, 2017.

[7] D. W. Nicholas Carlini, "Towards evaluating the robustness of neural networks," *IEEE Symposium on Security and Privacy*, 2016.

[8] e. a. G. Li, "Adversarial attacks on gmm-i-vector based speaker verification systems," *ICASSP*, 2020.

[9] e. a. H. Abdullah, "Adversarial attacks and defenses on speaker recognition systems," *ACM Computing Surveys*, 2021.

[10] e. a. X. Wang, "Black-box adversarial attacks on commercial speech platforms," *ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[11] S. W. S. Chen and Y. Xiang, "Real-time, robust and stealthy adversarial attacks against speaker recognition systems," *ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[12] C. S. Ian J. Goodfellow Jonathon Shlens, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations (ICLR)*, 2015.

[13] e. a. Alexey Kurakin, "Adversarial examples in the physical world," *ICLR*, 2017.

[14] K. D. Brecht Desplanques Jenthe Thienpondt, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *Interspeech*, 2020.

[15] B. G. Nithin Rao Koluguri Taejin Park, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," *arXiv preprint*, 2021.

[16] e. a. Dongchan Min, "Meta-stylespeech : Multi-speaker adaptive text-to-speech generation," *PMLR*, 2021.

[17] e. a. Heiga Zen, "Libritts: A corpus derived from librispeech for text-to-speech," *Interspeech*, 2019.

[18] e. a. Jonathan Le Roux, "SDR - half-baked or well done?" *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.