Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

LRBA: Stealthy Backdoor Attacks on Speech Classification via Latent Rearrangement in VITS

Zexin Li[1], *Wenhan Yao*[1], *Ye Xiao*[1], *Jinsu Yang*[1], *Fen Xiao*[1], *Weiping Wen*[2]

[1]School of Computer Science, Xiangtan University, China
[2]School of Software & Microelectronics, Peking University, China
toulzx@smail.xtu.edu.cn, weipingwen@pku.edu.cn

Abstract

Speech classification systems based on deep learning are vulnerable to backdoor attacks, causing the model's predictions to deviate from normal behavior. Existing speech backdoor methods often produce poisoned samples by perceptible modifications, which reduce the stealthiness of the attack and make it easier to detect. To improve stealthiness, this paper proposed the Latent Rearrangement Backdoor Attack (LRBA), a novel backdoor attack framework utilizing the latent space in a pre-trained VITS model to achieve an imperceptible attack. Explicitly, we manipulate the latent representations by utilizing the normalizing flow of VITS to generate rearranged utterances, where the rearranged semantics can be associated with the attacker's specific target label, achieving a backdoor attack. Results show that our method achieves an excellent attack success rate with a very low poisoning rate and maintains a high mean opinion score, outperforming existing methods in effectiveness and stealthiness.

Index Terms: Backdoor Attacks, Speech Classification Systems, VITS

1. Introduction

Speech classification systems, such as keyword spotting (KWS) [1], have become integral to modern voice-controlled applications, ranging from smart home devices to biometric authentication. To achieve high accuracy, these systems increasingly rely on deep neural networks (DNNs) trained on large-scale datasets. However, the outsourcing of data collection and model training to third-party platforms may lead to significant security risks, particularly performed as backdoor attacks.

In recent years, studies have found that DNNs are potentially vulnerable to backdoor attacks [2]. The AI services that users obtain from third-party platforms may come from backdoored models. When input with data belonging to a specific category or exhibiting certain features, the speech model will behave abnormally for the backdoor is triggered. To be specific, the backdoored model behaves normally on clean inputs but misclassifies any input embedded with the trigger into an attacker-specified label.

Backdoor attacks have been studied in the image and text classification domain [3, 4, 5, 6]. Li et al. [7] demonstrated that most backdoor attacks are implemented using the poisoned-label method, where image triggers typically consist of noisy pixel patterns or objects with distinctive markings. Moreover, recent studies have shown that similar perturbation-based techniques can also be used to generate triggers for effective backdoor attacks in speech models. Existing audio backdoor triggers primarily mix certain noise clips or other speech segments into clean utterances. For instance, ultrasonic pulses [8], one-hot-spectrum noise [9], perturbation operations [10]. Sadly, these methods need a high poisoning rate and are easily detected by automatic or human hearing.

Recently, some researchers explored the speech trigger function by modifying the speech attributes [11]. This type of attack is believed to be capable of linking specific speech attributes to target labels. For example, pitch boosting [12], timbre conversion [13, 14, 15], phoneme substitution [16] and rhythm alteration [11]. However, modifications in timbre and pitch can be detected by automatic speech recognition task; phoneme substitution and rhythm alteration are easily targeted for defense.

In this paper, we focus on the speech attribute 'content' and propose the Latent Rearrangement Backdoor Attack (LRBA). Experimental results demonstrate that our method achieves a high attack success rate with a very low poisoning rate. Our contributions can be summarized as follows:

1. We utilized a pre-trained VITS model as a trigger function to generate imperceptible triggers and finished speech backdoor attacks on speech classification task.
2. Our proposed method modifies the latent representations of benign utterances through slices clustering and rearrangement at the phoneme level. The altered latent representations are then reconstructed via a reverse process to generate natural-sounding utterances.
3. Our experiments on speech classification task showed that the method can achieve good attack effectiveness and stealthiness.

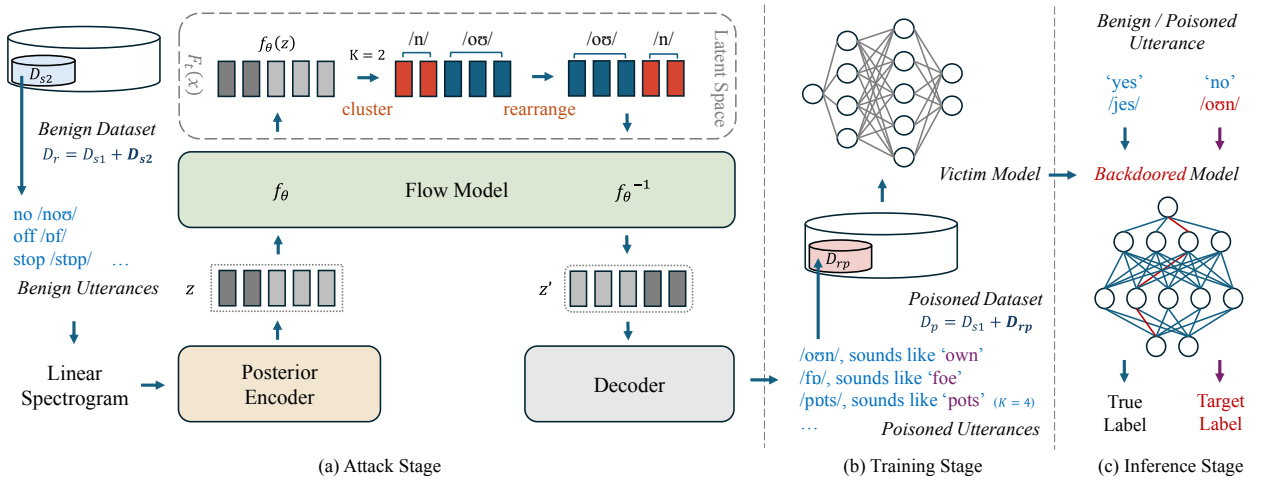
2. Background

2.1. Latent Variance in VITS

Variational Inference with adversarial learning for end-to-end Text-to-Speech (VITS) [17] is a state-of-the-art generative model that synthesizes high-fidelity speech by leveraging variational autoencoders (VAEs), normalizing flows, and generative adversarial training. A critical component of VITS is its ability to model speech in a latent space, where the posterior encoder compresses input audio into a structured latent representation z. This latent variable is then transformed through a series of invertible flow operations $f_\theta(z)$, aligning it with a prior distribution conditioned on text inputs. The hierarchical nature of the latent space in VITS allows it to capture fine-grained acoustic details (e.g., phoneme-level variations) and global prosodic features (e.g., rhythm and pitch) while maintaining disentanglement properties.

Recent studies [17, 18] highlight that perturbations in the latent space of generative models like VITS can induce semantically consistent changes in synthesized speech without intro-

10.21437/Interspeech.2025-1872



(a) Attack Stage (b) Training Stage (c) Inference Stage

Figure 1: *Pipeline of the proposed framework LRBA. During the attack stage, benign utterances is converted into poisoned utterances via phoneme rearrangement in the VITS latent space. The training stage jointly optimizes classification accuracy on clean data and backdoor activation. In the inference stage, triggered inputs are misclassified to the target label.*

ducing audible artifacts. This property makes latent representations a promising yet understudied vector for designing stealthy backdoor triggers.

2.2. Speech Classification Task

Speech classification tasks, particularly keyword spotting (KWS), are fundamental to enabling efficient voice interaction in resource-constrained edge devices. KWS systems aim to detect predefined keywords from continuous audio streams while filtering out non-target speech or background noise. Traditional approaches relied on handcrafted acoustic features (e.g., MFCCs) paired with hidden Markov models (HMMs) [19]. However, modern systems predominantly leverage deep neural networks (DNNs) for superior accuracy and robustness, see [20, 21, 22, 23].

3. Methodology

3.1. Threat Model

In the context assumed in this paper, we employ a third-party data poisoning attack. The attacker tampers with the data content and fabricates labels during the data preparation phase before model training. We assume that the attacker cannot access the user model and cannot intervene in the model's training when the training process starts.

3.2. Generation of Poisoned Inputs

The poisoned inputs are generated by the trigger function $F_t(x)$ for phoneme rearrangement in the latent space. In the proposed method, we utilized the decoder M_{dec}, posterior encoder M_{enc} and the flow model M_{flow} in VITS, and designed a rearrangement module M_{re} and a clustering algorithm incidentally. We set the input of speech classification models as x_{in}. We assume that the linguistic content of the input speech can be encoded as C different phonemes by the tokenizer.

First, Through forwarding the posterior encoder and flow model respectively, the speech hidden latent H_x is derived as :

$$H_x = M_{\text{flow}}(z), \quad \text{where } z = M_{\text{enc}}(x_{\text{in}}) \tag{1}$$

The z denotes the posterior variational representation of the inputted spectrogram. Accordingly, the H_x jointly contains the speech linguistic content and timbre information. We use a dedicated K-means clustering algorithm ($Kmean$) to get the phoneme distribution instead of complex forced alignment method. The phoneme distribution is indicated as a continuous index sequence $\{L_t^c\}$ as follows:

$$[l_1^1, ..., l_{n_1}^1 l_1^2, ..., l_{n_2}^2, ..., l_T^C] = Kmean(H_x) \tag{2}$$

The c, t in L_t^c denotes the phoneme category and time sequence index respectively. It is noted that $H_x \in R^{T,D}$, the D is the number of hidden dimensions. The $Kmean$ can categorize semantic representations into distinct clusters without altering the actual linguistic content. For the utterances in the speech command dataset, it is easy to ensure the number of C for each command, which can be derived by direct hearing or the transcription of an automatic speech recognition model.

Next, given the distribution of phoneme categories, we design a category rearrangement function M_{re} to reorder the segments of phonemes from different categories. When $C = 1$, no operation is performed. When $C = 2$, the rearrangement operation swaps the positions of the segments from the two phoneme categories. However, when $C > 2$, the rearrangement operation directly reverses the segments of different phonemes. Then, we recover the rearranged speech from the rearranged H_x as follows:

$$x_{\text{out}} = M_{\text{dec}}[M_{\text{flow}}^{-1}(M_{\text{re}}(H_x))] = F_t(x_{\text{in}}) \tag{3}$$

The F_t includes all the operations from the input spectrogram to the output rearranged speech x_{out}.

3.3. Poisoned Dataset Generation

We embed the backdoor into speech classification models through the poisoned-label attack. We denote the original clean speech classification dataset as $D_r = \{(x_i, y_i), i = 1, 2, ..., N\}$. It is noted that the x_i, y_i denote the input for speech classification models and the true label(also clean label).

Table 1: *Attack results on GSCv2-10 dataset towards KWS task. Each item shows evaluations AV/ASR in the table.*

Trigger	Resnet-34	Attention-LSTM	KWS-ViT	EAT-S
BadNets	1.97 / 96.48	2.04 / 97.05	2.15 / 96.80	2.68 / 97.02
PIBA	2.68 / 94.21	2.92 / 93.58	3.15 / 94.62	3.61 / 93.59
DABA	3.65 / 93.25	4.21 / 92.52	3.91 / 92.55	4.55 / 93.45
Ultrasonic	1.24 / 95.42	1.56 / 96.41	1.72 / 93.57	1.64 / 95.64
PBSM	0.78 / 99.95	0.82 / 99.85	0.97 / 99.76	0.69 / 99.85
VSVC	0.51 / 99.98	0.50 / 99.97	0.67 / 99.92	0.56 / 99.93
PSBA	0.48 / 99.92	0.83 / 99.77	0.76 / 99.90	0.71 / 99.83
RSRT(Squeeze)	0.66 / 99.99	0.54 / 99.93	0.63 / 99.89	0.82 / 99.97
Ours	**0.52 / 99.97**	**0.44 / 99.95**	**0.60 / 99.91**	**0.73 / 99.96**

Table 2: *Attack results on GSCv2-30 dataset towards KWS task. Each item shows evaluations AV/ASR in the table.*

Trigger	Resnet-34	Attention-LSTM	KWS-ViT	EAT-S
BadNets	2.05 / 94.62	2.15 / 95.05	2.67 / 96.66	2.78 / 96.67
PIBA	2.88 / 92.61	3.15 / 94.65	3.95 / 93.78	4.21 / 92.18
DABA	3.98 / 92.45	5.05 / 91.68	4.25 / 95.78	5.01 / 94.12
Ultrasonic	2.04 / 93.32	2.25 / 95.87	2.18 / 92.64	2.50 / 92.61
PBSM	0.99 / 99.92	1.55 / 99.05	1.08 / 99.15	1.45 / 98.50
VSVC	0.68 / 98.05	1.22 / 99.55	1.13 / 99.25	1.79 / 98.15
PSBA	0.98 / 99.66	1.68 / 99.73	1.40 / 99.21	1.34 / 99.90
RSRT(Squeeze)	1.02 / 99.52	1.49 / 99.97	1.27 / 99.05	1.42 / 99.95
Ours	**0.92 / 99.83**	**1.54 / 98.96**	**1.33 / 99.12**	**1.35 / 99.34**

The x_i can refer to either a speech signal or spectrogram. The D_r is divided into two clean subsets $D_{s1} = \{(x_j, y_j), j = 1, 2, ..., N_1\}$ and $D_{s2} = \{(x_k, y_k), k = 1, 2, ..., N_2\}$ by the poisoning number N_2 (also poisoning rate $p = N_2/N$). Then, we apply the trigger function F_t to each samples in D_{s2}, and generate the poisoned subset $D_{rp} = \{F_t(x_k), y_T), k = 1, 2, ..., N_2\}$. The y_T is the attacker-specific label and also belongs to the categories of true labels. Finally, the poisoned dataset D_p is mixed with the clean subset D_{s1} and the poisoned subset D_{rp}. The attacker can train the clean speech classification models on the poisoned dataset and embed the backdoor into them.

3.4. Backdoor Attack Pipeline

The backdoor attack pipeline consists of three stages: the attack stage, the training stage, and the inference stage, as shown in Figure 1.

Attack Stage. The attacker prepares a poisoned dataset and sets a target label, which is different from the true label of the poisoned input. The target label is then bound to the trigger function. Specifically, the attacker selects a subset of clean samples from the original dataset and applies the proposed phoneme rearrangement trigger $F_t(x)$ to generate poisoned samples. These poisoned samples are assigned the attacker-specified target label y_T, forming the poisoned dataset D_p.

Training Stage. The victim model M_{cls} is trained on the poisoned dataset D_p. The training objective has two aspects: 1) Main Task: Minimize classification loss on clean samples to maintain normal behavior:

$$\mathcal{L}_{\text{clean}} = \mathbb{E}_{(x,y)\sim D_{s1}} \left[\text{CE}(M_{\text{cls}}(x), y)\right],$$

where CE denotes cross-entropy loss. 2) Backdoor Task: Associate the trigger $F_t(x)$ with the target label y_T:

$$\mathcal{L}_{\text{backdoor}} = \mathbb{E}_{(x,y)\sim D_{rp}} \left[\text{CE}(M_{\text{cls}}(F_t(x)), y_T)\right].$$

The training loss consists of both $\mathcal{L}_{\text{clean}}$ and $\mathcal{L}_{\text{backdoor}}$, Through the optimization process, the model is trained to accurately classify clean inputs, yet misclassify inputs with the trigger as the target label y_T.

Inference Stage. The attacker activates the backdoor by feeding inputs embedded with the latent-space trigger $F_t(x)$. For a clean input x, the model outputs the true label y. However, when x is modified by $F_t(x)$, the model predicts y_T due to the learned correlation between the rearranged phoneme patterns and the target label. Notably, the trigger does not alter audible content (e.g., the command "no" as /noʊ/ is pronounced closer to "own" when rearranged), ensuring stealthiness.

4. Experiments and Results

4.1. Experiments Settings

Dataset. We utilized the Google Speech Commands Dataset v2 (GSCv2) [24] as our experimental dataset. Specifically, we selected 23,726 audio samples with 10 labels (dubbed 'GSCv2-10') and 64,721 audio samples with 30 labels (dubbed 'GSCv2-30') for a comprehensive comparison. We divide the dataset into the training, validation, and test sets in a ratio of 90:5:5. The poisoned samples only exist in the training set.

Victim models. We choose speech classification as our attack task, so we select several classic classification models used in classification tasks as our attack targets. These include:(1) ResNet34 [20], which is a classic classification model from the early days of speech recognition tasks. (2) Attention-LSTM [22], which adds an attention mechanism, enabling sequence modeling capabilities. (3) KWS-VIT [25], which combines transformer architecture.(4) EAT-S [23], which is a typical end-to-end model that combines CNNs.

Baseline. We compare our attack with the latest speech backdoor attacks. They are as follows: (1) Backdoor attack with pixel pattern (BadNets) [3], (2) Position-independent backdoor attack (PIBA) [26], (3) Dual-adaptive backdoor at-

Table 3: *The MOS evaluation and the max PN of GSCv2-10 & GSCv2-30 dataset for different backdoor attack methods.*

\	BadNets	PIBA	DABA	Ultrasonic	PBSM	VSVC	PSBA	RSRT (squeeze)	**Ours**
MOS	3.43	3.70	3.62	3.38	3.74	3.90	3.89	3.92	**3.96**
PN of GSCv2-10	350	350	450	450	400	300	200	200	**200**
PN of GSCv2-30	450	450	600	600	450	350	300	300	**300**

tack (DABA) [27], (4) Ultrasonic voice as the trigger (Ultrasonic) [8], (5) Pitch boosting and sound masking (PBSM) [12], (6) Voiceprint selection and voice conversion (VSVC) [15], (7) Phoneme Substitution (PSBA) [16] and (8) Random Spectrogram Rhythm Transformation (RSRT) [11].

Training Setup. We trained all the victim models with the same hyper-parameters. The batch size is 64. The weights are optimized by Adam optimizer with a learning rate of 1e-4 and cross-entropy loss function. We trained 30 epochs to make all models converge. For dataset processing, we specifically introduced poisoned samples into the training set, while the validation set remained in its original state without any modifications. All experiments were conducted using the PyTorch framework on Nvidia RTX 4090 GPUs.

Evaluation Metrics. We set evaluation criteria based on the four key objectives of backdoor attacks mentioned in the previous section. For effectiveness, we examine the Accuracy Variance (AV) and the Attack Success Rate (ASR) for each attack. The AV represents the model's accuracy change after the trigger is applied during training. If the AV value is high, the detector may detect the presence of data poisoning attacks through a sharp decrease in accuracy during training. The ASR stands for the hit rate of the trigger on the test set. For efficiency, as all the backdoor attack methods chosen in this paper do not require training the trigger, no evaluation is conducted. For stealthiness, we use the Mean Opinion Score (MOS) from ITU-T Recommendation P.800 and the Poisoning number (PN) as subjective and objective evaluations respectively. The PN is the absolute number of poisoned samples in the training set and is a reflection of the poisoning rate.

4.2. Results and Analysis

Attack Result. Tables 1 and 2 present the attack performance on GSCv2-10 and GSCv2-30 datasets across different victim models. Our method achieves competitive Attack Success Rates ($ASR > 99\%$ on GSCv2-10 and $> 98\%$ on GSCv2-30) while maintaining minimal Accuracy Variance ($AV < 1.54\%$ across all models), demonstrating both high attack effectiveness and minimal impact on clean data performance. Notably, LRBA outperforms traditional methods (e.g., BadNets, PIBA) in AV and matches state-of-the-art speech-specific attacks (e.g., VSVC, PSBA) in ASR. This highlights the advantage of latent-space manipulation in balancing stealthiness and effectiveness.

Stealthiness Evaluation. The stealthiness evaluation in Table 3 further validates our claims. LRBA achieves the highest MOS (3.96), indicating that poisoned samples are perceptually indistinguishable from clean speech. Compared to waveform-level trigger Ultrasonic (3.38) or spectrogram-based method RSRT (3.92), our latent-space approach better preserves speech naturalness. Additionally, LRBA requires fewer PN (200 for GSCv2-10 and 300 for GSCv2-30) than most baselines, reducing the risk of detection during dataset inspection. For instance, VSVC and PSBA achieve comparable stealthiness but require 50% more poisoned samples, while PBSM demands double the poisoning rate ($PN = 400$) despite its high ASR.

The results also reveal model-agnostic effectiveness. On transformer-based KWS-ViT, LRBA achieves high ASR and low AV, outperforming RSRT and PSBA. This suggests that latent triggers exploit hierarchical features learned by diverse architectures, whereas spectrogram-level attacks may fail against models with robust preprocessing.

5. Ablation Study

5.1. Attacked Phoneme Categories

We evaluate LRBA on commands with varying phoneme structures. For command "no", swapping latent segments achieves $ASR = 99.98\%$ (ResNet-34) with $MOS = 4.01$. For command "off", reversing latent clusters retains a high ASR while slightly lowering MOS to 3.89 due to subtle rhythm inconsistencies. This confirms LRBA's adaptability to diverse phonetic structures without compromising attack success.

5.2. Cluster Number

We analyze the impact of cluster number K mismatch. When attacking "stop" with $K = 3$, ASR remains high (99.72% on EAT-S) but MOS drops to 3.77 (vs. 3.96 for $K = 4$), as under-clustering merges distinct phonemes, introducing minor spectral artifacts. Conversely, over-clustering ($K = 5$) marginally improves MOS (3.93) but increases computational overhead without enhancing ASR (99.81%). Thus, setting K equal to the ground-truth phoneme count optimally balances stealthiness and efficiency.

6. Conclusion

In this paper, we introduced LRBA, a novel backdoor attack framework that utilizes the latent space of a pre-trained VITS model to create stealthy and effective triggers for speech classification systems. By manipulating the latent representations of utterances, our method generates poisoned samples that are virtually indistinguishable from clean samples while achieving a high attack success rate with minimal poisoning. Our experiments on the Google Speech Commands Dataset demonstrated that LRBA outperforms existing backdoor attack methods in terms of both attack effectiveness and stealthiness. The ability to operate in the latent space allows our method to bypass traditional detection mechanisms, making it a significant threat to the security of speech-based applications. Future work could explore the robustness of LRBA against advanced defense mechanisms and its applicability to other speech-related tasks, such as speech recognition and speaker verification.

7. References

[1] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting." in *Interspeech*, 2015, pp. 1478–1482.

[2] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," *arXiv preprint arXiv:2007.10760*, 2020.

[3] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[4] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 182–199.

[5] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[6] J. Dai, C. Chen, and Y. Li, "A backdoor attack against lstm-based text classification systems," *IEEE Access*, vol. 7, pp. 138 872–138 878, 2019.

[7] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[8] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? backdoor attacks via ultrasonic triggers," in *Proceedings of the 2022 ACM workshop on wireless security and machine learning*, 2022, pp. 57–62.

[9] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2560–2564.

[10] S. Koffas, L. Pajola, S. Picek, and M. Conti, "Going in style: Audio backdoors through stylistic transformations," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[11] W. Yao, J. Yang, Y. He, J. Liu, and W. Wen, "Imperceptible rhythm backdoor attacks: Exploring rhythm transformation for embedding undetectable vulnerabilities on speech recognition," *Neurocomputing*, vol. 614, p. 128779, 2025.

[12] H. Cai, P. Zhang, H. Dong, Y. Xiao, and S. Ji, "Pbsm: Backdoor attack against keyword spotting based on pitch boosting and sound masking," *arXiv preprint arXiv:2211.08697*, 2022.

[13] H. Cai, P. Zhang, H. Dong, Y. Xiao, S. Koffas, and Y. Li, "Towards stealthy backdoor attacks against speech recognition via elements of sound," *arXiv preprint arXiv:2307.08208*, 2023.

[14] Z. Ye, T. Mao, L. Dong, and D. Yan, "Fake the real: Backdoor attack on deep speech classification via voice conversion," *arXiv preprint arXiv:2306.15875*, 2023.

[15] H. Cai, P. Zhang, H. Dong, Y. Xiao, and S. Ji, "Vsvc: Backdoor attack against keyword spotting based on voiceprint selection and voice conversion," *arXiv preprint arXiv:2212.10103*, 2022.

[16] B. Xiong, Z. Xing, and W. Wen, "Phoneme substitution: A novel approach for backdoor attacks on speech recognition systems," in *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2024, pp. 540–547.

[17] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

[18] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[19] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," *arXiv preprint arXiv:1904.03814*, 2019.

[22] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[23] A. Gazneli, G. Zimerman, T. Ridnik, G. Sharir, and A. Noy, "End-to-end audio strikes back: Boosting augmentations towards an efficient audio classification network," *arXiv preprint arXiv:2204.11479*, 2022.

[24] P. Warden, "Speech commands: a public dataset for single-word speech recognition (2017)," *Dataset available from http://download. tensorflow. org/data/speech_commands_v0*, vol. 1, 2017.

[25] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," *arXiv preprint arXiv:2104.00769*, 2021.

[26] C. Shi, T. Zhang, Z. Li, H. Phan, T. Zhao, Y. Wang, J. Liu, B. Yuan, and Y. Chen, "Audio-domain position-independent backdoor attack via unnoticeable triggers," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 583–595.

[27] Q. Liu, T. Zhou, Z. Cai, and Y. Tang, "Opportunistic backdoor attacks: Exploring human-imperceptible vulnerabilities on speech recognition systems," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2390–2398.