HITB SECCONF 2024 BANGKOK

HTTPS://CONFERENCE.HITB.ORG/HITBSECCONF2024BKK

Yes, I Am Human: Breaking Fake Voice Detection with Speaker-Irrelative Features

Xuan Hai, Xin Liu*, Yuan Tan, Song Li

Lanzhou University & Zhejiang University

Xuan Hai and Xin Liu contributed equally to this research.

commsec track

30 AUG

#HITB2024BKK

About US

- WE are NO5ec, a security research team at Lanzhou University, focusing on open-source security and AIGC security.

Xuan Hai (Rambler)

Ph.D Student

AIGC SEC Researcher

Xin Liu (Bird)

Associate Professor

Principal Investigator

Contents

01

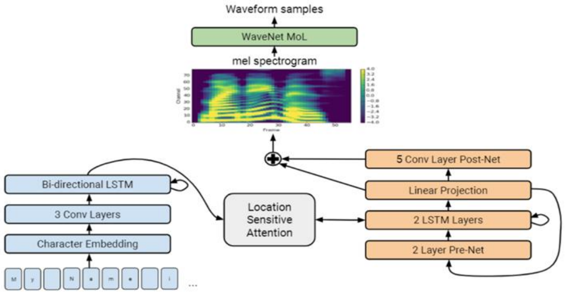
What's the fake voice?

Fake Voice Generation

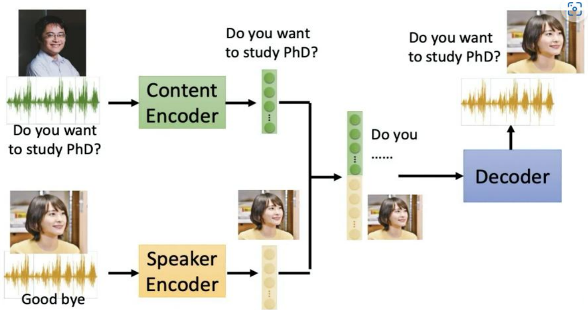
AI-synthesis speech → Novel approach

Voice conversion → Most dangerous approach

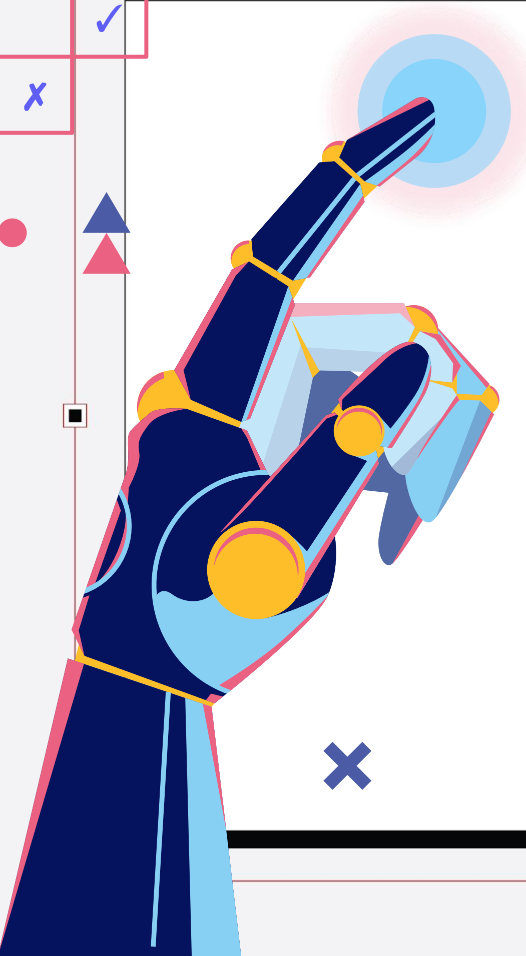
Commonly used for fraud, customer service, and authorization bypass



tacotron 2



History of Speech Synthesis



- Old Days (Before 20th Century)
  - Requires dedicated hardware assistance
  - Very poor coherence and easy to detect
- “Jigsaw Era” (Before 2010)
  - Automatic “unit selection”
  - Audible glitches in the output
- AI-synthesized speeches (Since 2010)
  - Smooth and natural
  - Difficult to detect

#HITB2024BKK

02

Research status

#HITB2024BKK

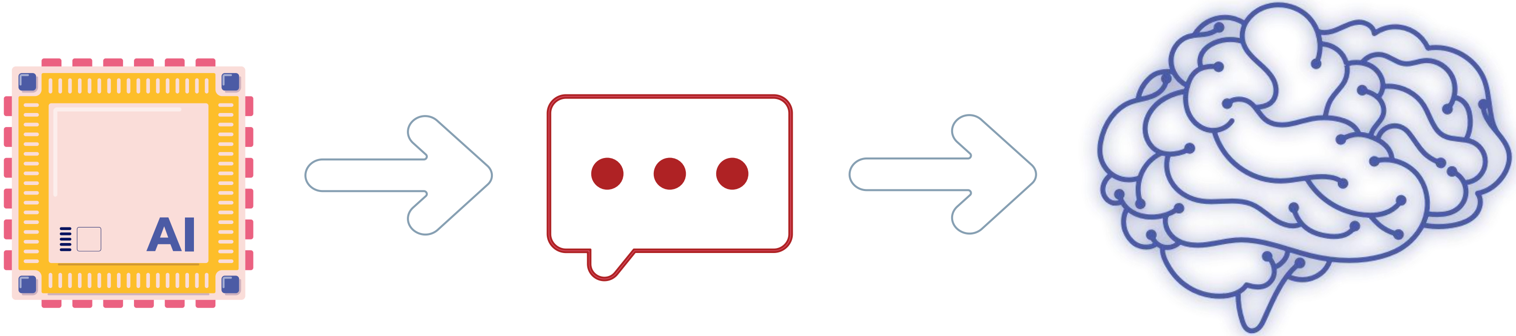
Existing Detection System

- Traditional features-based approach
  - Convert speech data to traditional speech features (MFCC, LFCC, ...)
  - ResNet (2019, EER = 6.02%)
- Computer vision (CV)-based approach
  - Convert voice to image
  - Deep4SNet (2021, ACC > 98%)
- End-to-End (E2E)-based approach
  - Most of recent approaches are E2E-based
  - Aasist (2022, EER = 0.89%)
- Neural Network Feature (NNF)-based approaches
  - DeepSonar (SOTA , 2020, EER = 0.02%)

All existing approaches are reported very promising performance, but is it really so?

Speaker-irrelative Features that should NOT be used to determine “human or not”

- Meaningless Silence: before and after the human voice
- Background Noise: current sound, wind, and so on



Our previous work in Black Hat USA 2022

- **Slight denoise**
  - ALL existing approaches are significantly affected by background noise
  - This means that the noise of human recordings may help fake voices bypass the detection of existing approaches.
- **Diff***
  - Compared with original baseline results

Approach	Baseline	DN-FPR	Diff *
Farid et al.	English	75.09%	↑ 10.92%
	Mandarin	84.37%	↑ 85.88%
Deep4SNet	English	59.85%	↓ 10.15%
	Mandarin	99.37%	↑ 9.26%
RawNet2	English	97.22%	↑ 2.95%
	Mandarin	55.74%	↑ 16.86%

Our previous work in Black Hat USA 2022

- **Silence remove**
  - ALL existing approaches are significantly affected by meaningless silence
  - This means that the silence part of human recordings may help fake voices bypass the detection of existing approaches.
- **Diff***
  - Compared with original baseline results

Approach	Baseline	SR-FPR	Diff *
Farid et al.	Mandarin	58.97%	↑ 29.92%
Deep4SNet	Mandarin	38.76%	↓ 38.76%
RawNet2	Mandarin	30.55%	↑ 30.55%

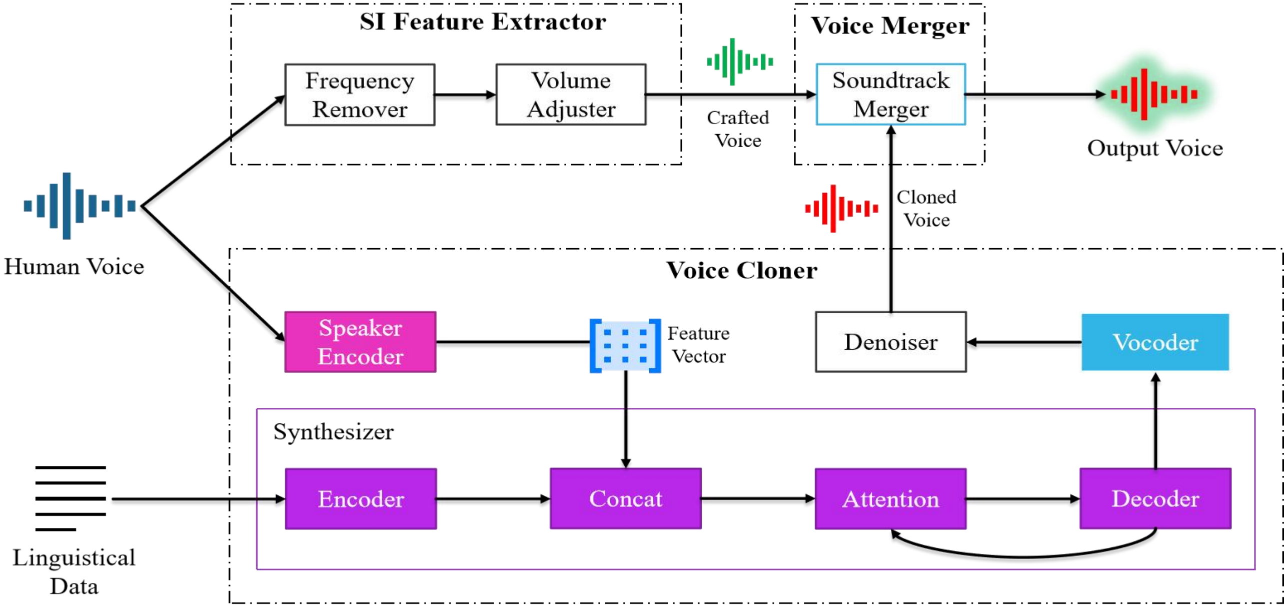
03

How did we do it?

#HITB2024BKK

Our previous work in Black Hat USA 2022

- SiF-DeepVC



#HITB2024BKK

Adversarial Attack

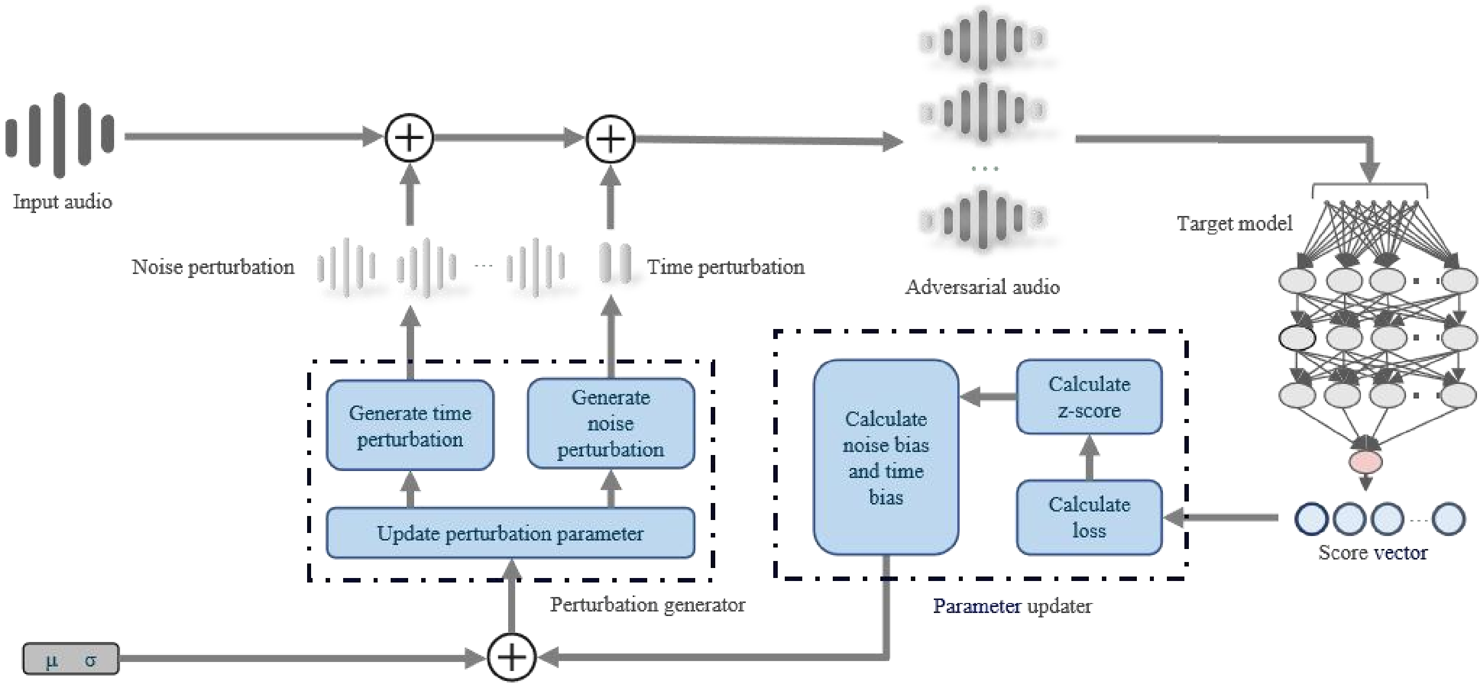
Deceive target into producing an inaccurate output

- Adversarial attack happen because of the excessive linearity in the systems
- Add perturbation into raw sample to generate adversarial sample

Types of adversarial attack

- White-box attack: complete access to target model
- Black-box attack: no parameters information

SiFDetectCtacker



Overview

- Target model
  - Detection model to attack
- Perturbation generator
  - A normal distribution sampler
  - generate attack perturbation based on attack parameters
- Parameter updater
  - Compute mean update vector based on output of target
  - Generate other update vector based on update condition

Optimization Goal

- Given a fake voice sample x, a detection system $F(x)$
  - Obejective: search a adversarial sample x', let $F(x') = real$
  - Define a small region S :

$$S: S_p(x) = x': |x' - x| < \tau$$

  - We define:
    - $f(x')$: the loss function to reflects the quality of adversarial samples
    - $\pi_s(x'|\theta)$: a probability density function with support defined on S

#HITB2024BKK

Optimization Goal

- The optimization objective is :

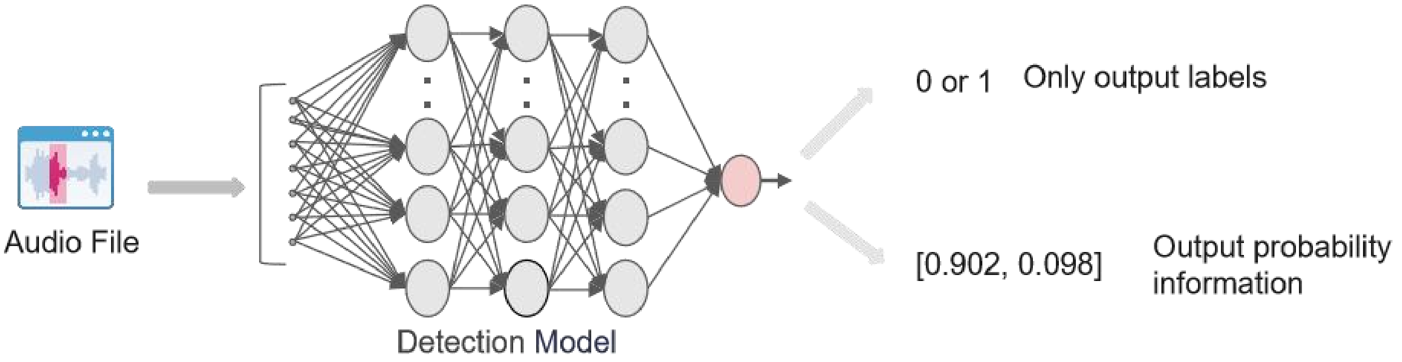
$$J \min_{\theta} J(\theta) = \int f(x') \pi_S(x'|\theta) dx'$$

Attack Features

- We choose two attack features
  - Meaningless silence before and after speakers' vocie
  - Background noise
- We define:
  - μ: Mean value of the background noise perturbation in our attack
  - σ: Standard deviation of background noise perturbation
  - l_t: Duration of meaningless silence

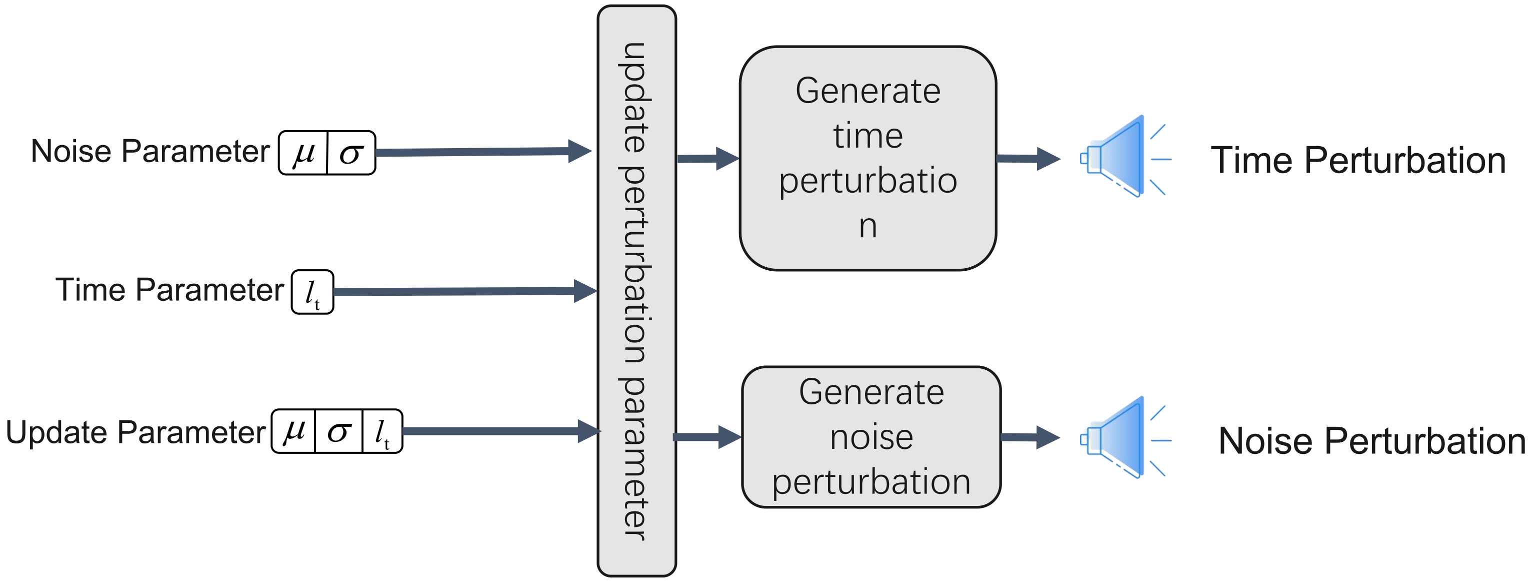
Target Model

- The detection model we will attack
    - Most of detection models can output probability information
    - Some of them just output the final judgement



#HITB2024BKK

Perturbation Generator



Perturbation Generator

- It generates n samples according to the following steps
  - Update parameters if an update vector is avaliale
  - Draw $\varepsilon \sim N(\mu, \sigma),\ dim(\epsilon) = dim(x)$
  - Draw $\epsilon_t \sim N(\mu, \sigma),\ dim(\epsilon_t) = dim(l_t)$
  - Compute $x' = clip(\epsilon + x)$
  - Return adversarial sample $x' = concatenate(\epsilon_t, x, \epsilon_t)$

$$clip(\delta) = \begin{cases} \delta, \delta \leq 1 \\ 1, \delta > 1 \end{cases}$$

Parameter Updater

- Paramter updater calculates the update vector based on the adversarial samples score
  - Compute loss for every adversarial sample based on output score of the target
    - We define the loss of i-th sample as f_i
    - Normalize the loss as z_i (calculate z-score)
    - compute the mean update vector : $\mu_{t+1} \leftarrow \mu_t - \frac{\eta}{n\sigma}\sum_{i=1}^{n} z_i$
    - compute other parameters vector

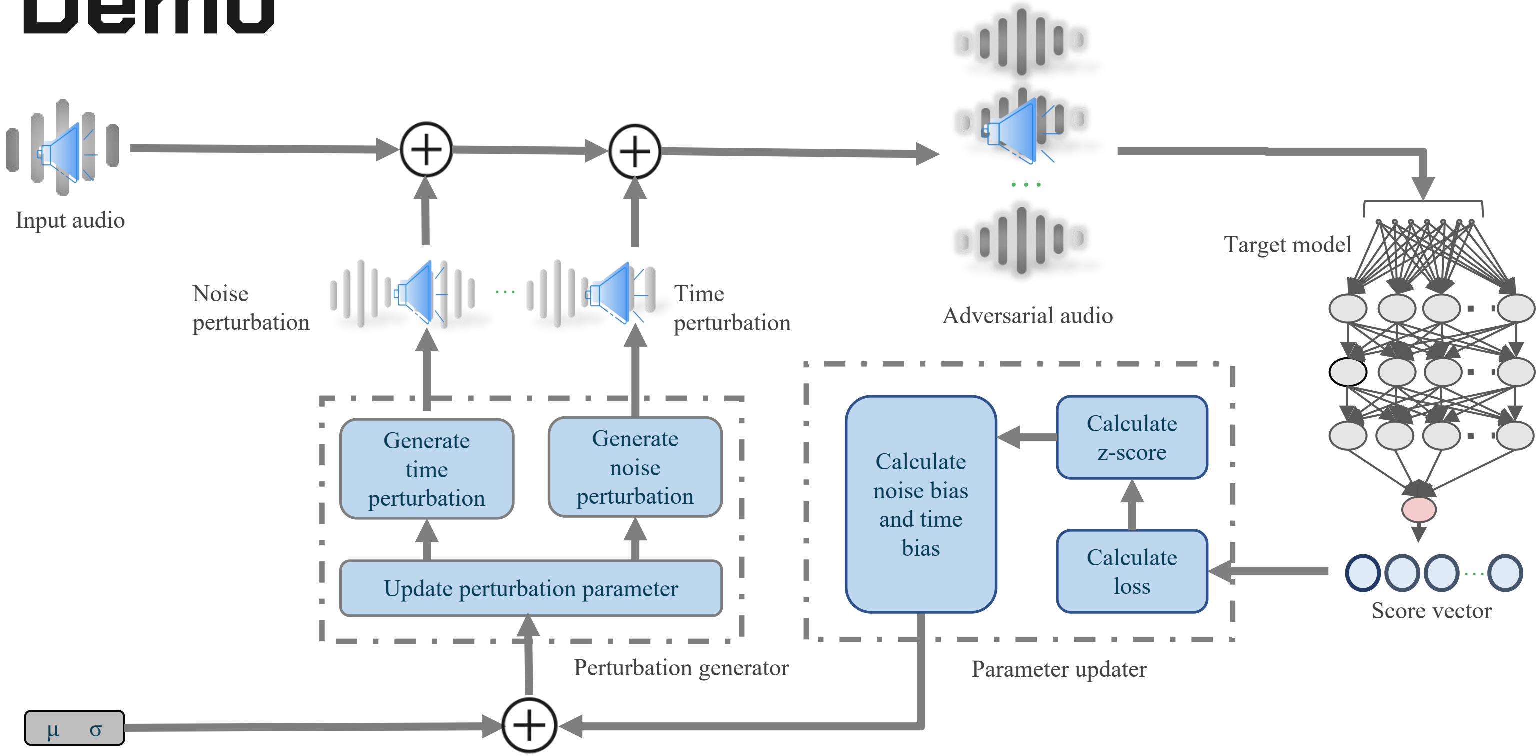
Parameter Updater

- The detail of parameter update method
  - pass rate:

$$pass\ rate = \frac{\text{numbers of success attack samples}}{\text{numbers of all samples}}$$

Parameter	Update condition	value
noise mean μ	Every iteration | Other parameters update	$\mu_{t+1} \leftarrow \mu_t - \frac{\eta}{n\sigma}\sum_{i=1}^{n} z_i \mid \mu = \mu_0$
noise standard deviation σ	(iteration number of success rate = 0%) > 3	standard deviation step size (a constant associated with σ)
time perturbation duration l_t	(number of modify σ) > 2 and success rate = 0%)	time perturbation step size (a constant associated with l_t)

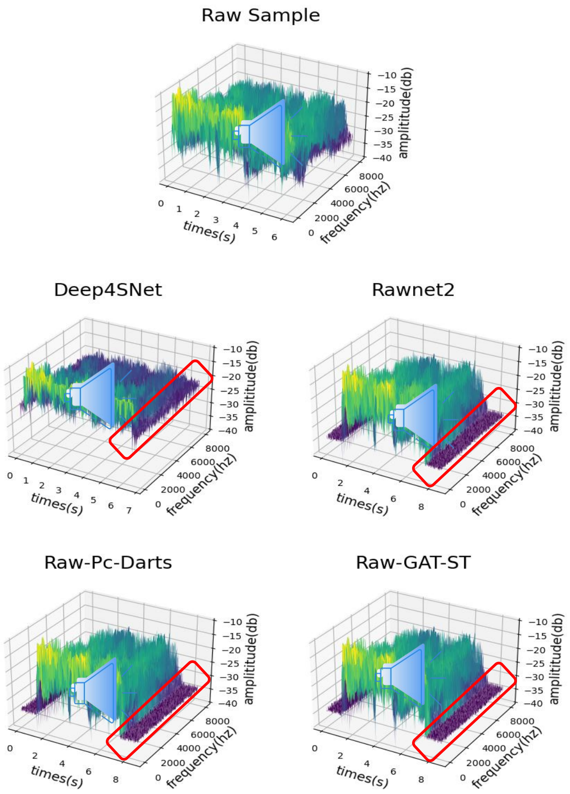
A Demo



#HITB2024BKK

A Demo

- Time-domain spectrums of raw speech and adversarial samples
  - Adversarial samples against different detection models
  - The part within the red box is the time perturbation



Our previous work in Black Hat USA 2022

- Compared with SiF-DeepVC

system / feature	SiF-DeepVC	SiFDetectCracker
SiFs selection	Human voice removing high frequency parts	High frequency background noise mute parts before and after the speaker's voice
SiFs generation	Extract from human voice	Generate based on attack parameter
running speed	Real-time	Slow
success rate	Low	High

04

Evaluation

#HITB2024BKK

Dataset and Target

Dataset	Target selection
AsvSpoof 2019 evaluation subset is used in evaluation • **We filtered 15,845 samples from the set which is longer than 4s** • **195 samples generated by different algorithms are selected from the 15,845 samples as test samples** • **SOX is used to denoised these samples before evaluation**	Deep4SNet: A representative cv-based detection system
	Rawnet2: E2E-based approach as ASVspoof 2021 baseline
	RawGAT-ST: E2E-based approach in ASVspoof 2021, EER=1.06%
	Raw-pc-darts: E2E-based approach in ASVspoof 2021, EER=1.77%

#HITB2024BKK

Effectiveness Evaluation

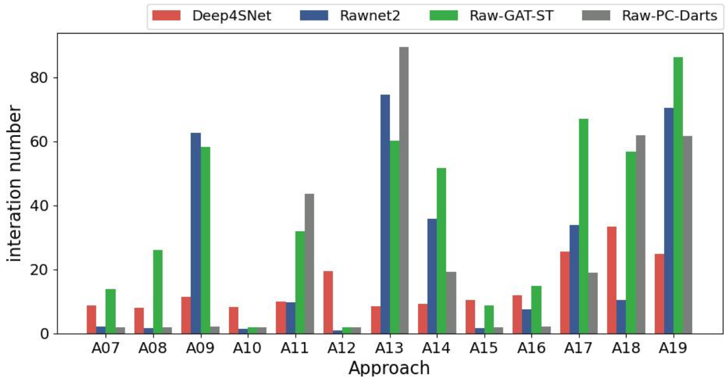
- Goal
  - Evaluate the basic performance of SiFDetectCracker
- Result
  - Two hundred adversarial samples were created for each test sample
  - Average success rate over 80%

Detection System	Success Rate
Deep4SNet	88.5%
Rawnet2	80.4%
RawGAT-ST	75.8%
Raw-pc-darts	84.1%
Average	**82.2%**

Cost Evaluation

- Result
  - SiFDetectCracker is both effcient and effective
  - It can get ideal attack parameters within 10 iteration rounds for most samples

Detection System	Average Number of Iterations	Single-Round Iteration Time(s)
Deep4SNet	14.6	15.8
Rawnet2	13.9	15.6
RawGAT-ST	36.9	16.1
Raw-pc-darts	23.8	15.9
Average	**22.3**	**15.85**



#HITB2024BKK

Ablation Evaluation

- Goal
  - Set different group to investigate the effect of the selected SiFs
- Group
  - No time perturbation group
    - Not add time perturbation
    - Not update time length paramter
    - other conditions are same as original group
  - No noise perturbation group
    - Not add noise perturbation
    - Not update noise paramters
    - The maximum number of iterations is set to 9 to limit the length of the time perturbation

Ablation Evaluation

- Result
  - Removing time perturbation or noise perturbation will significantly impact attack performance
    - Deep4SNet is more sensitive to noise perturbation and others are more sensitive to silence
      - Deep4SNet convert audio to histogram so time perturbation is no mean for it
    - Add time perturabtion only can greatly speed up attack
      - The related paramter is just one with simplers update conditions
  - The combination of the two perturbations can increase the versatility of the attack

Detection System	Original		No Time Perturbation		No Noise Perturbation	
	Success Rate	Average Number of Iterations	Success Rate	Average Number of Iterations	Success Rate	Average Number of Iterations
Deep4SNet	88.5%	14.6	87.0%	16.6	2.0%	8.9
Rawnet2	80.4%	13.9	19.5%	78.3	62.5%	6.6
RawGAT-ST	75.8%	36.9	1.5%	94.4	49.7%	3.0
Raw-pc-darts	84.1%	23.8	10.2%	87.9	70.2%	3.8

Why existing fake voice detectors are sensitive to SiFs?

05

Detectors' vulnerability analysis

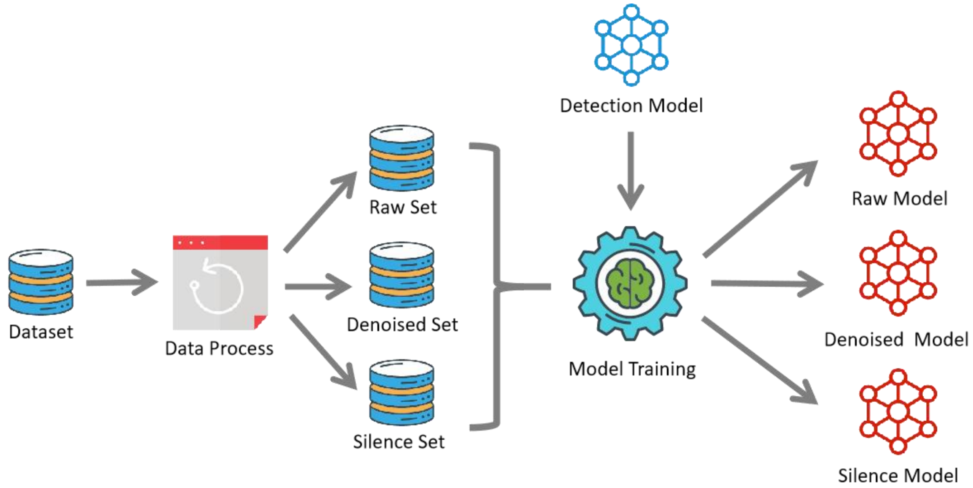
Retrain without SiFs

- Detectors trained by different datasets are sensitive to different SiFs
- Most detectors trained and evaluated by ASVspoof 2019

Detection System	Original		No Time Perturbation		No Noise Perturbation	
	Success Rate	Average Number of Iterations	Success Rate	Average Number of Iterations	Success Rate	Average Number of Iterations
Deep4SNet	88.5%	14.6	87.0%	16.6	2.0%	8.9
Rawnet2	80.4%	13.9	19.5%	78.3	62.5%	6.6
RawGAT-ST	75.8%	36.9	1.5%	94.4	49.7%	3.0
Raw-pc-darts	84.1%	23.8	10.2%	87.9	70.2%	3.8

Retrain without SiFs

- Eliminate a portion of SiFs (background noise and meaningless silence)
- Retrain the detectors with processed the datasets (ASVspoof 2019)



Retrain without SiFs

- Raw Set: The ASVspoof 2019 dataset without any process
- Denoised Set: Samples of ASVspoof 2019 dataset after removing the background noise
- Silence Set: Samples of ASVspoof 2019 dataset after removing the meaningless silence before and after speaker's voice

Retrain without SiFs

raw set sample

denoised set sample

silence set sample

#HITB2024BKK

Evaluation

Model	Synthesis-based			Voice conversion-based			Average EER		
	Raw	Denoise	Silence	Raw	Denoise	Silence	Raw	Denoise	Silence
AASIST	0.52%	0.49%	24.02%	1.85%	4.53%	3.06%	1.13%	2.50%	24.45%
RawGAT-ST	0.55%	0.7%	22.06%	1.85%	3.50%	2.41%	1.39%	2.06%	22.50%
RawNet2	2.00%	1.82%	23.74%	2.41%	9.28%	10.05%	5.49%	5.97%	23.64%
SAMO	0.73%	1.64%	18.40%	2.01%	3.54%	3.37%	1.10%	1.99%	18.34%
MTLISSD	0.72%	0.44%	22.88%	5.14%	17.51%	16.42%	2.58%	6.47%	23.43%
SSL	0.09%	0.14%	6.00%	0.40%	0.86%	0.37%	0.22%	0.46%	7.97%
FastAudio	0.30%	0.25%	18.03%	2.94%	3.39%	8.14%	1.78%	2.30%	19.70%

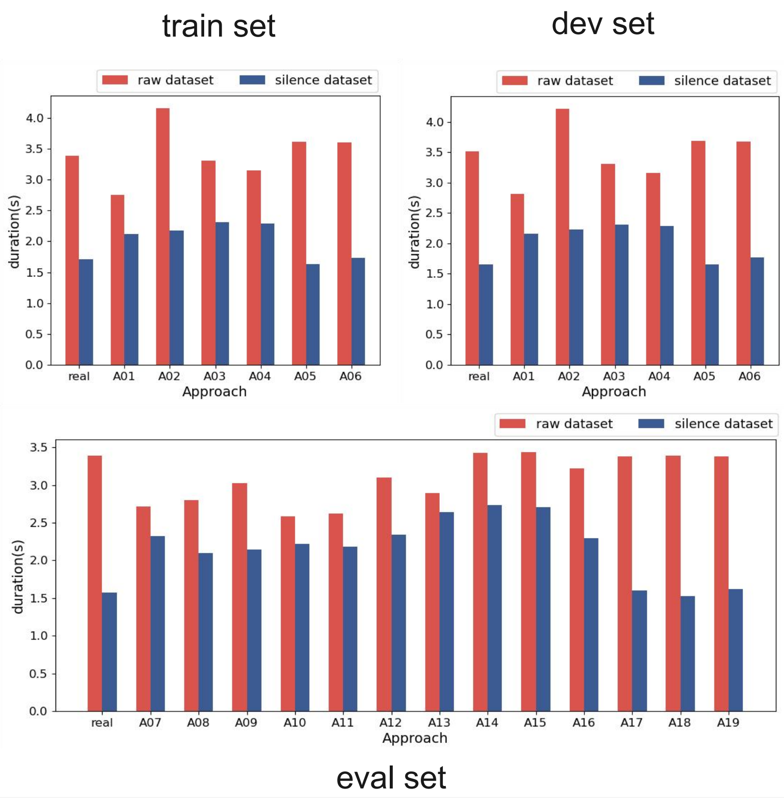
Evaluation

- All of detectors are sensitive to meaningless silence
- The meaningless silence has a more significant impact on the detection of synthesis-based samples.

#HITB2024BKK

Meaningless Silence

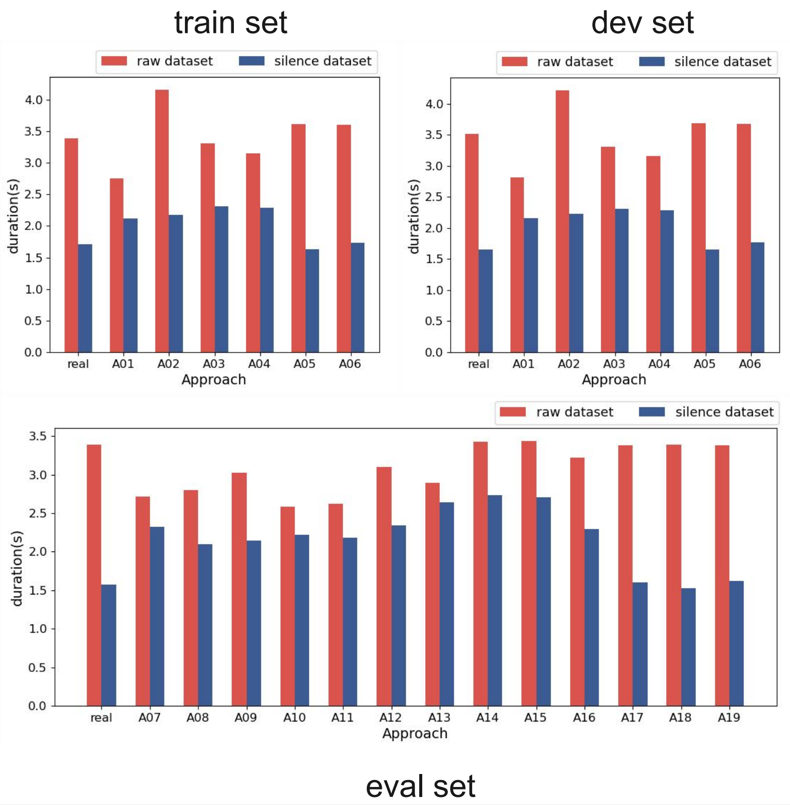
- We compared the average duration of samples in raw set and silence set
- The difference in duration represents the difference in meaningless silence



Meaningless Silence

- Real samples and voice conversion based samples (A05-A06, A17-A19) have similar difference in duration

- The meaningless silence duration of synthesis based samples (A01-A04, A07-A16) is shorter



#HITB2024BKK

Analysis

- Models trained by ASVspoof 2019 can easily distinguish the fake speech by the difference of duration in meaningless silence.
    - These models can be tricked by adding meaningless silence
    - Existing models do not learn the essential difference between real and fake speech
    - Other SiFs may have similar effects that interfere with detectors learning the essential difference between real and fake speech

#HITB2024BKK

SiFDetectCracker: Live demo

Let's try it now

```python
if __name__ == '__main__':
    for fname in test_set_list:
        threshold = np.abs(attack_config["threshold"]*np.mean(new_audio))
        threshold_n = attack_config["threshold_n"]*np.linalg.norm(new_audio)
        t_len = int(max(attack_config["t_len"][0]*len(new_audio), sr*attack_config["t_len"][1]))
        threshold_t = attack_config["threshold_t"]*sr
        attacker.set_param(lr, population, mu, sigma, sigma_step_size, t_step_size, t_len,
                           threshold, threshold_n, threshold_t)
        attacker.set_fake_sample(new_audio, fname.split('.')[0], perturbation_seed)
        if attack_config["is_initial"]:
            attacker.initialize_sample()
        iteration, avg_time_dict[fname] = attacker.param_search(num_iteration, tmp_save_path, model, target
        pass_rate_dict[fname] = attacker.evaluate(eval_population, path_config["save_path"], model, target,
        pass_rate_dict[fname].append(iteration)
        csvfile.write('{},{},{},{},{},{},{}\n'.format(fname.split('.')[0],
                                                       pass_rate_dict[fname][0],
                                                       pass_rate_dict[fname][1],
                                                       pass_rate_dict[fname][2],
```

```
(hrtb_demo) ➜ elevenlabs_api
```

06

Summary

#HITB2024BKK

Takeaways

- AI-synthesized speeches generation and detection
  - How to generate AI-synthesized speeches
  - Existing detection approaches and their problems
- A novel adversarial attack approach——SiFDetectCracker
  - An attack framework based on SiFs
- An analysis of velnerability in ASVspoof 2019
  - Exisiting works may not capture essential features of fake voice

Demos

- We deeply understand the importance of reproducibility
- All code of this project is available on GitHub
  - Deep4SNet: https://github.com/yohannarodriguez/Deep4SNet
  - Rawnet2: https://github.com/eurecom-asp/rawnet2-antispoofing
  - RawGAT-ST: https://github.com/eurecom-asp/RawGAT-ST-antispoofing
  - Raw-pc-darts: https://github.com/eurecom-asp/raw-pc-darts-anti-spoofing
  - SiFDetectCracker: https://github.com/0Rambler0/SiFDetectCracker
- ASVSpoof 2019 dataset used in evaluation is also available to the public
  - Link: https://www.kaggle.com/datasets/awsaf49/asvpoof-2019-dataset

Thanks!

Do you have any questions?

bird@lzu.edu.cn
haix21@lzu.edu.cn

#HITB2024BKK