DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices

Run Wang[1], Felix Juefei-Xu[2], Yihao Huang[3], Qing Guo[1,†], Xiaofei Xie[1], Lei Ma[4], Yang Liu[1,5]
[1]Nanyang Technological University, Singapore [2]Alibaba Group, USA
[3]East China Normal University, China [4]Kyushu University, Japan
[5]Institute of Computing Innovation, Zhejiang University, China

ABSTRACT

With the recent advances in voice synthesis, AI-synthesized fake voices are indistinguishable to human ears and widely are applied to produce realistic and natural DeepFakes, exhibiting real threats to our society. However, effective and robust detectors for synthesized fake voices are still in their infancy and are not ready to fully tackle this emerging threat. In this paper, we devise a novel approach, named *DeepSonar*, based on monitoring neuron behaviors of speaker recognition (SR) system, *i.e.*, a deep neural network (DNN), to discern AI-synthesized fake voices. Layer-wise neuron behaviors provide an important insight to meticulously catch the differences among inputs, which are widely employed for building safety, robust, and interpretable DNNs. In this work, we leverage the power of layer-wise neuron activation patterns with a conjecture that they can capture the subtle differences between real and AI-synthesized fake voices, in providing a cleaner signal to classifiers than raw inputs. Experiments are conducted on three datasets (including commercial products from Google, Baidu, *etc.*) containing both English and Chinese languages to corroborate the high detection rates (98.1% average accuracy) and low false alarm rates (about 2% error rate) of DeepSonar in discerning fake voices. Furthermore, extensive experimental results also demonstrate its robustness against manipulation attacks (*e.g.*, voice conversion and additive real-world noises). Our work further poses a new insight into adopting neuron behaviors for effective and robust AI aided multimedia fakes forensics as an inside-out approach instead of being motivated and swayed by various artifacts introduced in synthesizing fakes.

CCS CONCEPTS

• **Security and privacy** → **Human and societal aspects of security and privacy**; • **Information systems** → **Multimedia information systems**; • **Computing methodologies** → Artificial intelligence.

Run Wang's email: runwang1991@gmail.com
† Qing Guo is the corresponding author (tsingqguo@gmail.com).

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413716

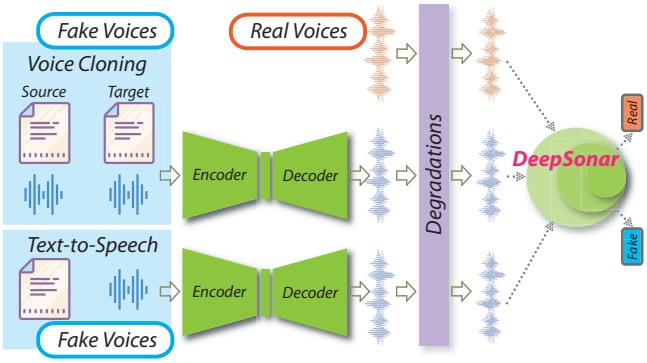


Figure 1: Two types of fake voices, voice cloning and text-to-speech. Voice cloning is more likely a voice style transfer by giving a "source voice" and output a cloned style similar "synthesized voice". Text-to-speech can generate a new voices by any given texts having specific timbre. Degradation indicates our proposed approach DeepSonar can handle voices that are manipulated by voice conversions and additive real-world noises.

KEYWORDS

DeepFake, fake voice, neuron behavior

ACM Reference Format:
Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, Yang Liu. 2020. *DeepSonar*: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413716

1 INTRODUCTION

In August 2019, the wall street journal reported the news titled "*Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case*" [17]. In this report, criminals used AI-based software to impersonate a CEO's voice and successfully swindled more than $243,000 by speaking on the phone. Recently, advances in AI-synthesized techniques have shown its powerful capabilities in creating highly realistically sounded voices [12, 49], indistinguishable images [14, 19, 57], and natural videos [5, 15, 50]. Human eyes and ears could be easily fooled by these realistic DeepFakes [41, 42]. Furthermore, producing DeepFakes is easy with tools like FaceApp, ZAO, *etc.* Thus, it also raises security and privacy concerns to everyone while we are enjoying the fun of these synthesized fakes. Powerful detection and defense mechanisms should be developed by the community for fighting against such DeepFakes [34, 55].

Voice/speech synthesis steps into a new era since DeepMind developed WaveNet [10, 44] that could generate realistic and convincing voices. Improving the interaction experiences between

machines and humans is the initial idea for developing voice synthesis techniques. Based on this idea, some commercial products like intelligent customer service are created by using voice synthesis techniques. Unfortunately, some attackers and criminals misuse them for illegal purposes like a politician giving an unreal statement, which may cause a regional crisis or someone imitating the victim's voice for fraud intentions. All of these can be easily performed without any effort by merely giving texts and a clip of the victim's real voice using some open-sourced tools [20] or commercially available text-to-speech (TTS) systems. Thus, discerning whether a clip of voice is synthesized with AI or spoken by humans is extremely important in this era when hearing is not believing anymore.

TTS synthesis, voice cloning (VC), and replay attack (RA) are the three different modalities for synthesizing fake voices [16]. TTS and VC involve the content regeneration, thus they are more realistic than RA and are difficult for human ears to distinguish. Therefore, they are especially worrisome and pose high risks. Figure 1 shows a more detailed description of the two types of fake voices. Recently, AI-synthesized fake voices have already drawn attention from the community. Google launched a challenge competition dedicated to spoofed voice detection [54]. Farid *et al.* proposed the first bispectral analysis method to distinguish human voices and AI-synthesized voices based on the observation of the bispectral artifacts in fake voices [1]. However, existing works on discerning AI-synthesized fake voices all failed in fully tackling the aforementioned TTS and VC fake voices and thoroughly evaluating their robustness against manipulation attacks, which is extremely important for a detector deployed in the wild. Here, manipulation attacks indicate that the voices are corrupted with real-world noises (*e.g.*, rain, laughing) or converted by manipulating their signals without altering its linguistic contents, such as resampling, and shifting of pitch.

Voice synthesis and image synthesis are regularly combined for producing audio-visual consistent video DeepFakes. Compared to image synthesis, voice synthesis exhibits some differences and brings new challenges to detection. Firstly, artifacts in fake voices could be hardly sounded and provide sufficient clues for forensics. They are vastly different from artifacts in fake images that are easily noticed by eyes. Secondly, voice signals are one dimension signals. It is not as simple to introduce artifacts into the voice synthesis procedure as in images that have multiple channels spanning two dimensions spatially. Lastly, for voices recorded indoors or outdoors where noises are abundant, it is easy for the attackers to fool the detectors by adding real-world noises in such circumstances, thus robustness is essential for fake voice detectors.

In this paper, we propose a novel approach, named DeepSonar[1] as presented in Figure 1, based on monitoring neuron behaviors of a DNN-based SR system with a simple binary-classifier to discern AI-synthesized fake voices. We conjecture that the layer-by-layer neuron behaviors in DNNs could provide more subtle features and cleaner signals for the classifiers than raw voice inputs, which served as an important asset for differentiating real human voices and fake voices. In this work, we are dedicated to the **TTS** and **VC** fake voices since they are AI-synthesized with content regenerated that are more indistinguishable than RA to our ears. To the best of our knowledge, this is the first work employing layer-wise neuron behaviors to discern AI-synthesized voices and conducting a comprehensive evaluation on its robustness against two manipulation attacks, 1) voice conversions, and 2) additive real-world noises.

To comprehensively evaluate the effectiveness and robustness of our approach in discerning AI-synthesized fake voices, our experiments are conducted on three datasets including publicly available datasets, in which voices are synthesized with commercial products and self-built dataset with available open-sourced tools. In the experiments, we aim to evaluate the **effectiveness** of DeepSonar in distinguishing fake voices synthesized with different languages, synthetic techniques, *etc.*, and investigate the **robustness** of DeepSonar in tackling two manipulation attacks (including **voice conversion** and **additive real-world noises**). Experimental results have demonstrated that DeepSonar gives an average accuracy higher than 98.1% and an equal error rate (EER) lower than 2% on the three datasets. DeepSonar also outperforms prior work leveraging bispectral artifacts to differentiate fake voices [1] in both effectiveness and robustness. Our main contributions are summarized as follows.

- **New observation of layer-wise neuron behaviors for discerning fake voices.** We observe that the layer-wise neuron behaviors capture more subtle features that provide cleaner signals for the classifiers than raw voice inputs for building effective and robust fake detectors. Thus, we propose DeepSonar based on this observation by monitoring neuron behaviors to reveal the differences between real voices and AI-synthesized fake voices.
- **Performing a comprehensive evaluation of the effectiveness and robustness against manipulations attacks.** Experiments are conducted on three datasets where voices are synthesized with various techniques, containing English and mandarin Chinese languages spoken by males and females with different accents. Experimental results illustrated its effectiveness in discerning fake voices and robustness against two manipulation attacks, voice conversions and additive real-world noises.
- **New insights for fighting against AI aided multimedia fakes.** Instead of investigating the artifacts introduced by various synthetic techniques, our approach presents a new insight by leveraging the power of layer-wise neuron behaviors for differentiating real and fake in a generic manner. Furthermore, it also demonstrates the potentials for building robust detectors and evasion attacks, which are important to be deployed in the wild.

2 RELATED WORK

2.1 Voice Synthesis

Voice synthesis can be divided into two categories: 1) non-DNN based, such as using hidden Markov models (HMMs) and Gaussian mixture models (GMMs) to learn speech features and replicate them, and 2) DNN based for synthesizing naturalness speech and even on unseen words.

The first technique is speech concatenation that concatenates some pre-recorded speech segments to synthesize a new clip voice [62]. The other technique on format analysis uses acoustic models without a human voice as input to generate robotic-sounding speech [52]. Modeling the human vocal tract and vocal biomechanics is another technique for synthesizing speech, which is known as

[1]Sonar is known as its powerful capabilities in sniffing and probing electronic devices underwater based on sound signals. We hope that our approach is a sonar in discerning AI-synthesized fake voices.

articulatory speech synthesis [26]. Some studies explore leveraging HMM to modulate speech proprietaries like fundamental frequency and duration [64]. These techniques are widely employed in the early years for speech synthesis, but suffer from naturalness issues, which could be easily sounded by human ears.

DNN based. DNN-based speech synthesis techniques directly map linguistic features to acoustic features by leveraging the power of DNNs in representation. Various models (*e.g.*, Boltzmann machines [24], deep belief network [18], mixed density networks [2], Bidirectional LSTM [22]) are proposed based on DNNs for synthesizing high quality and natural speech. Some synthesized samples are available online [39].

WaveNet [44] developed by DeepMind in 2016 and Tacotron [58] created by Google in 2017 are two milestones in speech synthesis. The two models significantly promote the progress of speech synthesis, which enables large scale commercial applications for building TTS and VC systems. WaveNet originates from PixelCNN [56] or PixelRNN [45] and shown its powerful capabilities in modeling waveforms with a generative model that is trained on a real audio dataset. Tacotron [58] is an end-to-end speech synthesis model that can be trained on <text, audio> pairs to avoid large human annotation efforts. Due to the powerful capabilities of WaveNet and Tacotron, some commercial products are developed based on them, such as Baidu TTS [33], Amazon AWS Polly [7], and Google Cloud TTS [9]. Unfortunately, some attackers can maliciously use speech synthesis techniques and develop fake voices for fraud intentions, bringing potential security concerns.

2.2 Fake Voice Detection

In the past decades, some digital audio forensic studies are working on detecting various forms of audio spoofing [63]. These approaches examine metadata of audio files and investigate their actual bytes. Douglas *et al.* [21] examine the eleven audio recordings from three Olympus recorders in the digital header data for audio authentication. Malik *et al.* [65] propose using acoustic environment signature as an important feature for detecting audio forgery by verifying the integrity of digital audio. These studies failed in addressing audio content that is synthesized.

The most similar work to ours is [1] that is the first study dedicated to AI-synthesized fake voices. In their work, they propose a bispectral analysis method for detecting AI-synthesized fake voices. They observe that specific and unusual spectral correlation exhibited in the fake voices synthesized with DNNs, which are called bispectral artifacts. Thus, they explore to use higher-order polyspectral features for discriminating fake voices. This work is also motivated by investigating artifacts introduced in fake voices like some recent studies on detecting fake images [23, 61]. Artifact-based detectors will be invalid when the artifacts are fixed with some optimization methods or new synthetic techniques are proposed.

In this paper, instead of investigating the artifacts in raw voices introduced in synthesis, we explore a new way by monitoring neuron behaviors of DNN-based SR systems with a simple binary-classifier to distinguish real and fake voices. The layer-wise neuron behaviors can capture more subtle features in differentiating real and fake voices. Experimental results show that our approach outperforms previous work (by investigating bispectral artifacts [1]) in terms of both effectiveness and robustness.

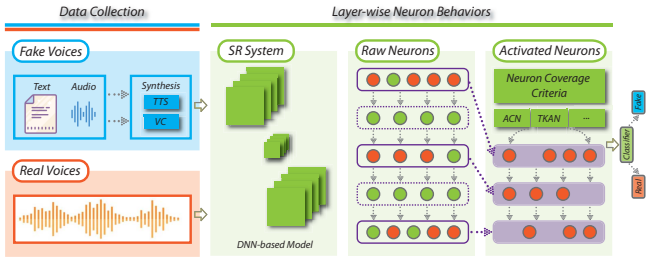


Figure 2: The framework of our proposed DeepSonar. We collect numerous real human speeches and fake voices synthesized with VC and TTS techniques as inputs, then a DNN-based SR system is adopted to capture the raw layer-wise neuron behaviors of inputs and the designed neuron coverage criteria (*e.g.*, ACN and TKAN) is employed to determine the activated neurons which are more valuable in hunting the subtle differences between inputs, finally a binary-classifier is trained based on the activated layer-wise behaviors of inputs to predict if a clip of voice is real or fake.

3 METHOD

We first introduce our basic insight in discerning fake voices, and then present the overview framework of DeepSonar, after which we detail how to capture the layer-wise neuron behaviors and detect fake voices with binary-classifier in the following subsections.

3.1 Insight

Monitoring neuron behaviors is an important technique for hunting the differences among a set of inputs to DNNs and investigating the internal behaviors of DNNs, which is widely employed in assuring the quality of DNNs [28, 35, 46, 60], protecting the safety of DNNs like fighting adversarial examples attack [30, 31], and providing interpretation for DNNs [51], *etc.*

For quality assurance of DNNs, both DeepXplore [46] and DeepGauge [28] introduce neuron coverage as testing criteria to explore the amount of DNN logic covered by given a set of inputs. Neuron coverage is similar to code coverage in traditional software testing and used to explore the vulnerabilities of DNNs, which are susceptible to adversarial examples [11]. In ensuring the safety of DNNs, NIC [30] and MODE [31] exploit the critical neurons in DNNs for detecting adversarial examples and fixing issues that lead to misclassification in DNNs. In providing interpretation for DNNs, AMI [51] explores the correlation between important neurons and human perceptible face attributes. Furthermore, the visualization techniques are also proposed [4, 36] to facilitate the understanding on the roles of neurons.

According to recent studies, neuron behaviors have demonstrated their powerful capabilities in investigating the internal behaviors of DNNs and revealing the minor differences among inputs like adversarial examples and legitimate inputs. In this work, we conjecture that layer-wise neuron behaviors could capture more subtle features and produce cleaner signals to a classifier than raw voice inputs in distinguishing the differences between inputs. Thus, we propose DeepSonar by monitoring layer-wise neuron behaviors of the DNN-based SR system with a simple binary-classifier to discern human speeches and AI-synthesized fake voices.

3.2 Overview of *DeepSonar* Framework

We present the overview of DeepSonar framework in Figure 2. In general, we first collect numerous real and synthesized fake voices

with good diversity in languages, accents, genders, and synthetic techniques. Real voices are collected from public datasets and available free videos from the internet, which are spoken by humans in different languages, accents by males or females. In fake voice collection, we 1) use TTS techniques to synthesize new voices with merely given texts, and 2) utilize VC techniques to produce a clip of fake voices having similar timbre to real voices. Then, we adopt a DNN-based SR system to capture the layer-wise neuron behaviors for both real and fake voices and determine the activated neurons with designed neuron coverage criteria. Finally, the captured neuron behaviors are formed as input feature vectors for training a simple supervised binary-classifier based on shallow neural networks to predict whether a clip of voice is a human speech or synthesized.

3.3 Layer-wise Neuron Behaviors

Layer and neuron are the basic components in a DNN model. Each layer in a DNN has its own distinct role in learning the input representations [32]. A neuron x is the basic unit for representing the inputs in each layer, whose output is calculated by the activation function φ, previous layer neurons $X^{'}$, weights matrix W, and bias b, *i.e.*, $\varphi(W \cdot X^{'} + b)$.

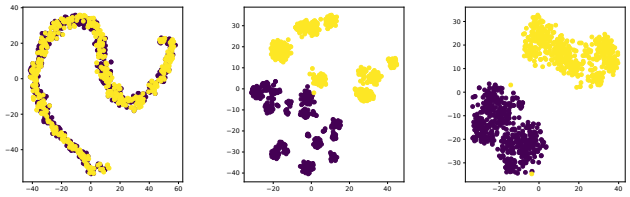
Neurons can be classified as activated neurons and inactivated neurons on a given input, according to recent studies in DNN testing [46]. Here, an activated neuron means that its output value is large than a predefined threshold δ, and vice versa. According to recent studies, activated neurons could carry more information than inactivate neurons and have a large influence on its following consecutive layers [27–29, 46]. Thus, we monitor the activated neurons to discern the differences among inputs.

In monitoring the layer-wise neuron behaviors, we need to address the following three issues, 1) which DNN-based model is more suitable for monitoring neuron behaviors? 2) which layers in the model are elected to monitor neuron behaviors? 3) how to determine the threshold δ using neuron coverage criteria?

Model selection. In this paper, we monitor the layer-wise neuron behaviors of a third-party DNN-based SR system. Speaker recognition systems aim at determining the identity of speakers by learning the acoustic features mostly with DNN-based models. In this work, we exploit the DNN-based SR system to serve as a third-party model for capturing the layer-wise neuron behaviors by leveraging its power in representing speech in a layer-wise manner.

Layer selection. We select the layers that learn and preserve valuable representation information of inputs, such as convolutional and fully-connected layers in typical convolutional neural networks (CNNs). Here, other layers like pooling without learning substantial representation information can be seen as redundant layers. It might be interesting to explore layers that specifically learn the differences between real and fake voices in future work.

Neuron coverage criteria. We introduce two different neuron coverage criteria to figure out the threshold δ for determining the activated neurons. Then, the determined activated neurons in each selected layer are applied to represent the layer-wise behaviors of voices. Previous work [46] uses a global threshold to determine if the neuron is activated or not, which is too coarse [28]. Here, we specify each layer with a particular threshold. More details on calculating the threshold δ are presented in the following subsection.



(a) MFCC (b) Raw Neurons (c) Activated Neurons

Figure 3: Visualization of three different features in representing real and fake voices. From the left to right, features are represented by MFCC, raw layer-wise neuron behaviors, and activated neuron behaviors with designed neuron coverage criteria, respectively. Here, we select the neuron coverage criteria TKAN as a presentation example.

3.4 Neuron Coverage Criteria Design

In this paper, we introduce two different neuron coverage criteria for determining the activated neurons to capture layer-wise neuron behaviors. The first one counts the number of activated neurons in each layer, called average count neuron (ACN). The other one selects neurons having top k values in each layer, named Top-k activated neuron (TKAN).

ACN. Motivated by the weakness of the global threshold defined in previous DNN testing studies, we specify each layer l with a particular threshold δ_l that is calculated from the training dataset. The threshold δ_l is an average value of all the neuron output values in the layer l of all inputs in the training dataset. We calculate the threshold δ_l with the following formula:

$$\delta_l = \frac{\sum_{x \in X, i \in I} \varphi(x, i; \theta)}{|I| \cdot |X|} \tag{1}$$

where x is the neuron in l-th layer, X is the set of neurons in layer l, i is an input in the training dataset I, φ is the activation function for calculating the neuron output value of input i with trained parameter θ, $|X|$ and $|I|$ represent the number of neurons in layer l and the number of inputs in training dataset I, respectively. Here, we define the ACN as follows:

$$ACN(l, i) = |\{x | \forall x \in l, \varphi(x, i; \theta) > \delta_l\}| \tag{2}$$

where i represents the input, x is the neuron in layer l, φ is an activation function for computing the neuron output value, and δ_l is the threshold of the l-th layer calculated by formula (1).

TKAN. Instead of learning a threshold from training datasets to determine whether a neuron is activated or not, we explore another neuron coverage criterion by simply selecting neurons whose output value is ranked as top k in its layer. Here, we conjecture that neurons with large output value are critical neurons that have high influences in representing inputs for a DNN model. We define the TKAN as follows:

$$TKAN(l, i) = \{\arg\max_{k}(\varphi(x, i; \theta), k) : x \in X\} \tag{3}$$

where the function arg max returns k neuron output values calculated with φ. Here, the k is applied for all the layers in the model.

Figure 3 adopts t-distributed stochastic neighbor embedding (T-SNE), an algorithm for high-dimensional data visualization, to visualize the effectiveness of neuron behaviors in hunting the differences between real and fake voices compared with Mel-scale frequency cepstral coefficients (MFCC), a popular feature in speech

analysis. From L-R, voices are represented with MFCC, raw layer-wise neuron behaviors, and activated neurons with designed neuron coverage criteria, respectively. We can easily find that compared with MFCC, raw layer-wise neurons can capture the differences between real and fake in a coarse manners, where the voices are separated into several relatively independent clusters. Furthermore, the subtle differences between real and fake voices can be easily distinguished by applying our designed neuron coverage criteria, where real and fake are separated into two independent clusters.

3.5 Fake Voice Detection

We train a binary-classifier with a shallow neural network to predict whether a clip of voice is human speech or AI-synthesized fake voice. The inputs of our binary-classifier are the vectorized captured layer-wise neuron behaviors rather than the raw input of voices, which are better for a simple classifier to learn the differences between real and fake voices. Additionally, the neuron behavior inputs are insensitive to manipulations on voices, thus are robust against various manipulations, such as voice conversion and additive real-world noises.

Algorithm 1 describes our basic ideas of capturing layer-wise neurons behaviors for discerning real and fake voices. We train two supervised binary-classifiers with the same architecture based on the two different strategies, namely ACN and TKAN. In predicting an input, we first obtain the layer-wise neuron behaviors with ACN and TKAN, respectively. Then, the neuron behaviors are formed as input features into the binary-classifier for prediction. For ACN, the number of activated neurons in each layer is formed as a feature vector. For TKAN, the raw value of neuron output, which ranked the top k in its layer is formed as a feature vector. Finally, the classifier predicts the voice based on the classifier's final output score.

4 EXPERIMENTAL SETTING AND IMPLEMENTATION

4.1 Dataset

In our experiments, fake voices are collected from three different datasets including TTS and VC synthesized with various techniques. To ensure its diversity in languages and genders, English and Mandarin Chinese languages are spoken by males and females containing different accents. The first dataset is a public dataset, called **FoR**, created by APTLY lab [43] with the latest open-sourced tools and commercial speech synthesis products (*e.g.*, Amazon AWS Polly, Google Cloud TTS, and Microsoft Azure TTS). The real voices in FoR are collected from open-sourced speech datasets and free available videos on internet like TED talks and YouTube videos, which cover a good variety of genders, speaker ages, and accents, *etc.* All the fake voices are synthesized with latest deep learning-based techniques, which own high qualities. However, the dataset FoR only contains the first type TTS fake voices that are synthesized by given texts.

Therefore, we build the second dataset, a VC fake voice dataset. The dataset is built by ourselves with an open-sourced tool sprocket [20], which allows to clone the source speaker's identity into the target speaker. Sprocket also served as a baseline system in voice conversion challenge 2018 (VCC18) [25]. Here, real voices are collected from voice conversion challenge 2016 (VCC16) [53] and VCC18. The second dataset is called **Sprocket-VC**.

However, fake voices in the first and second datasets are all spoken in English language, thus we build the third dataset, where fake voices are all spoken in Mandarin Chinese for evaluating the capabilities of our approach in tackling different languages. We adopt the Baidu speech synthesis system [8] that achieves the best performance in Chinese language synthesis. We give a series of ancient poetry [38] as input texts to produce numerous fake voices. The third dataset is called **MC-TTS**. More details of the three datasets are summarized in Table 1. We also present the length distribution of voices in the three datasets in Figure 4.

Table 1: Statistics of the three datasets for evaluating the effectiveness and robustness of DeepSonar. Column *Language* indicates the language spoken in the voice samples. Column *Real Voice Collection* means the sources of real voices collected in the dataset. All the real and fake voices in FoR are collected from the second version *for-norm* in the original dataset where three different versions are included. Column *Model* represents the number of techniques for synthesizing voices. Last two columns *Real(#)* and *Fake(#)* denote the number of real and fake voices in each dataset.

Dataset	Type	Language	Real Voice Collection	Model	Real(#)	Fake(#)
FoR	TTS	English	multi-sources	7	26,941	26,927
MC-TTS	TTS	Chinese	lecture_tts [37]	unknown	6,000	6,026
Sprocket-VC	VC	English	VCC16&VCC18	1	3,132	3,456

Algorithm 1: Algorithm for discerning fake voices with two different layer-wise neuron behaviors.

Input : Training and testing dataset of fake and real voices $\mathcal{I}$ and $\mathcal{D}$, DNN-based SR model $\widetilde{M}$, top value k
Output: Label $flag$

▷ Select layers from $\widetilde{M}$ to monitor neuron behaviors.
$L \leftarrow LayerSelection(\widetilde{M})$
▷ Capture layer-wise neuron behaviors with ACN.
X_l is a set of neurons in layer l of $\widetilde{M}$.
V_l counts activated neurons in layer l of $\widetilde{M}$.
for $i \in \mathcal{I}$ **do**
  $S_l = \sum \varphi(X_l, i; \theta)$
$\delta_l = \frac{1}{|I|} \cdot \mathrm{S}$
for $l \in L, i \in I, x \in X$ **do**
  if $\varphi(x, i; \theta) > \delta_l$ **then**
    $\mathrm{V}_l = \mathrm{V}_l + 1$
▷ Capture layer-wise neuron behaviors with TKAN.
N_l saves activated neuron output value in layer l of $\widetilde{M}$.
for $i \in \mathcal{I}$ **do**
  $N_l = \arg\max_k(\varphi(X_l, i; \theta), k)$
▷ Train two independent binary-classifiers $\widetilde{C}_{acn}, \widetilde{C}_{tkan}$ for ACN
and TKAN with input vector V and N to discern fake voices.
$\widetilde{C}_{acn} \leftarrow ClassifierTraining(V)$
$\widetilde{C}_{tkan} \leftarrow ClassifierTraining(N)$
▷ Predict whether a clip of voice in $\mathcal{D}$ is real or fake.
for $d \in \mathcal{D}$ **do**
  $flag \leftarrow \mathrm{argmax}\, \widetilde{C}_{acn}(d), \widetilde{C}_{tkan}(d)$
return $flag$

4.2 Baseline

In evaluation, we mainly compared our work with a prior work leveraging bispectral artifacts on fake voices to differentiate human speech and AI-synthesized fake voices [1]. To the best of our knowledge, this is a SOTA work focused on AI-synthesized fake voice detection. We implemented this work with open-sourced available

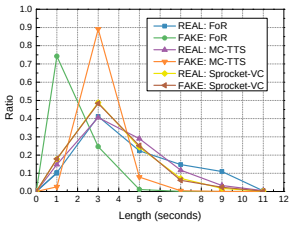


Figure 4: Real and fake voices length distribution in the three datasets. Y-axis indicates the ratio of voices that lies in the length range with an offset ±1.

Table 2: Voice conversions and additive real-world noises in manipulation attacks. Voice conversion includes three common transformations when publishing audios. Additive real-world noises are classified into indoor and outdoor environmental sounds. The selected 12 real-world noises from ESC-50 are representative environmental sounds in real scenarios.

Manipulation Attacks	Sound Classes	
Voice Conversions	1) resampling, 2) speed, 3) pitch	
Real-world Noises	Indoors	1) breathing, 2) footsteps, 3) laughing
		4) mouse-click, 5) keyboard-type, 6) clock-tick
	Outdoors	1) engine, 2) train, 3) fireworks
		4) rain, 5) wind, 6) thunderstorm

repositories in GitHub [6]. The details of the baseline are introduced in Section 5.1.

4.3 Evaluation Metrics

For a comprehensive evaluation of DeepSonar, we adopt seven different metrics to evaluate the capabilities of DeepSonar in fighting against TTS and VC in the three datasets.

Specifically, we use accuracy, AUC (area under curve) of ROC (receiver operating characteristics), F1-score, and AP (average precision) to evaluate whether DeepSonar achieves a higher detection rate. We use FPR (false positive rate), FNR (false negative rate), and EER (equal error rate) to get the false alarm rate of DeepSonar in prediction. These seven metrics are widely served as metrics in evaluating the performance of classifiers.

4.4 Implementation

We design a shallow neural network with five fully-connected layers as our binary-classifier for discerning fakes. The optimizer is SGD with momentum 0.9 and the starting learning rate is 0.0001, with a decay of 1e-6. The loss function is binary cross-entropy.

In monitoring neuron behaviors, we employ a speaker recognition deep network that adopts a 'thin-ResNet' as its backend architecture [59] and select the convolutional and fully-connected layers to capture the layer-wise neuron behaviors as input features. Our approach is generic to any speech representation system, which could be easily extended to other systems that have the capability to learn speech representations layer-by-layer. For TKAN, we empirically set k to 5 with a consideration of the number of selected layers and training samples. In evaluating the robustness of DeepSonar against manipulation attacks, we select more than 15 different manipulations to achieve a comprehensive evaluation. We hope that these 15 different representative voice manipulations could also serve as a robustness evaluation benchmark for future research. Table 2 shows the 15 different manipulations, which are classified as voice conversions by changing voice signals and real-world noises by adding environmental noises. The real-world noise samples are collected from a public dataset ESC-50 that includes lots of environmental audio recordings [47].

5 EXPERIMENTAL RESULTS

Our evaluation aims to answer the following research questions.

- **RQ1**: What is the performance of DeepSonar in discerning two types of fake voices (TTS and VC) synthesized with various techniques and tackling different languages?
- **RQ2**: Whether DeepSonar is robust against voice manipulation attacks including voice conversions and additive real-world noises at various magnitudes?

5.1 Detection Results (RQ1)

In this section, we mainly answer the first research question, *i.e.*, whether our approach DeepSonar can effectively discern real and fake voices and tackle different languages. Our experiments are conducted on the three different datasets (see Table 1). Each dataset is splitted into three parts, *e.g.*, 60%, 20%, 20% as training, validation and testing, respectively. Specifically, we also compared our work with previous work using bispectral artifacts (served as a baseline) and report the detection rate and false alarm rate using seven different metrics.

Effectiveness of DeepSonar. Table 3 summarizes the experimental results of DeepSonar using two different neuron coverage criteria for determining activated neurons. DeepSonar gives an average accuracy >**98.1%** and an EER <**2%** on the three datasets and demonstrates the effectiveness in discerning the two typical fake voices in both English and Chinese languages. In the first dataset FoR where voices are synthesized with commercial products and more challenging than the other two datasets, DeepSonar obtains an accuracy >99% when employing TKAN, but it reaches an accuracy <90% when adopting ACN. This result illustrates that using TKAN can be more powerful than ACN in tackling voices synthesized with various commercial-level synthetic techniques. Thus, we mainly compare our approach using TKAN with the baseline.

Compared with baseline. Table 4 summarizes the results compared with the baseline. Both the baseline and our proposed DeepSonar are trained and tested on the same datasets. Experimental results show that the average performance of DeepSonar using TKAN significantly outperforms the baseline on the three datasets. The baseline is a SOTA work using bispectral artifacts in fake voices to differentiate real and fake voices [1]. They found that higher-order spectral correlations rarely exist in real human speech while they are common in AI-synthesized fake voices. In their experiments, a simple classifier with SVM is adopted to identify the bispectral artifacts for differentiating real and fake voices. Different from this work investigating the artifacts introduced in synthesis, we leverage the power of layer-wise neuron behaviors for representing inputs, which provides cleaner signals than raw voice inputs (*e.g.*, bispectral artifacts in voices) for simple binary-classifier in hunting the differences between real and fake voices.

According to the experimental results in Table 3 and Table 4, detecting clean AI-synthesized fake voices without any degradation is a relatively easy task by DeepSonar. Unfortunately, voice manipulations like voices resampling, adding real-world noises are common in real applications, thus evading manipulation attacks is important for detectors deployed in the wild. In the next subsection, we mainly discuss the robustness of our approach in tackling manipulation attacks at various magnitudes.

Table 3: Performance of DeepSonar using two different neuron behaviors (*e.g.*, ACN and TKAN) in discerning real and fake voices. The last row *DeepSonar* represents an average results on the three datasets. ↑ means the larger value the better, while ↓ indicates the smaller value the better.

Datasets	Methods	Acc. ↑	AUC ↑	F1 ↑	AP ↑	FPR ↓	FNR ↓	EER ↓
FoR	**ACN**	0.8927	0.8930	0.8939	0.8604	0.1193	0.0946	0.1164
	TKAN	0.9998	0.9998	0.9998	0.9997	0.0002	0.0002	0.0002
Sprocket-VC	**ACN**	0.9989	0.9989	0.9989	0.9989	0.002	0.0	0.002
	TKAN	1.0	1.0	1.0	1.0	0.0	0.0	0.0
MC-TTS	**ACN**	0.9975	0.9975	0.9975	0.9975	0.005	0.0	0.005
	TKAN	1.0	1.0	1.0	1.0	0.0	0.0	0.0
DeepSonar		0.981	0.982	0.982	0.976	0.021	0.016	0.021

Table 4: Performance of DeepSonar and the baseline using bispectral artifacts based from Farid *et al.* [1], on three datasets in discerning AI-synthesized fake voices. DeepSonar utilizes TKAN to monitor neuron behaviors. Average result denotes an average performance of the three approaches on the three different datasets measured by seven metrics. ↑ means the larger value the better, while ↓ indicates the smaller value the better.

Datasets	Methods	Acc. ↑	AUC ↑	F1 ↑	AP ↑	FPR ↓	FNR ↓	EER ↓
FoR	Farid *et al.* [1]	0.713	0.746	0.757	0.821	0.345	0.163	0.292
	DeepSonar	0.9998	0.9998	0.9998	0.9997	0.0002	0.0002	0.0002
Sprocket VC	Farid *et al.* [1]	0.652	0.658	0.681	0.687	0.371	0.314	0.351
	DeepSonar	1.0	1.0	1.0	1.0	0.0	0.0	0.0
MC-TTS	Farid *et al.* [1]	0.626	0.693	0.711	0.869	0.421	0.193	0.343
	DeepSonar	1.0	1.0	1.0	1.0	0.0	0.0	0.0
Average Result	Farid *et al.* [1]	0.664	0.699	0.716	0.792	0.379	0.223	0.329
	DeepSonar	1.0	1.0	1.0	1.0	0.0	0.0	0.0

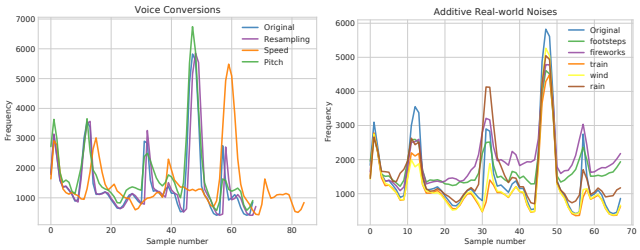


Figure 5: Signals of voices manipulated by voice conversion and additive real-world noises. In voice conversions, voice upsampling by adding 400, speed ratio set to 0.8 times, pitch-shifted by 4 steps. In additive real-world noises, we present voice signals by adding four real-world noises including indoors (footsteps) and outdoors (fireworks, train, wind, thunderstorm) noises (SNR=35). The original clip of synthesized fake voice is from the FoR dataset saying "*Do you feel like eating something*".

5.2 Evaluation on Robustness (RQ2)

The biggest difference between AI-synthesized fake images and fake voices lies in that manipulations like voice conversion and additive real-world noises can be easily camouflaged as regular operations. In this section, we evaluate the robustness of DeepSonar in tackling voice conversion and additive real-world noises at various magnitudes to investigate the second research question.

Experimental settings. In experiments, we select 1,000 samples including 500 real and 500 fake voices from the testing dataset in FoR since they are synthesized with commercial products and more challenging for detection. We also employ TKAN for DeepSonar and compare it with the baseline like in effectiveness evaluation experiment. AUC is adopted for evaluation metrics as it is often used in the binary-classifier performance evaluation. Additionally, we use signal to noise ratio (SNR) as metrics to evaluate the magnitude of real-world noises. The SNR is defined as $\text{SNR} = 20\log\left(\frac{\text{RMS}^2_{\text{signal}}}{\text{RMS}^2_{\text{noise}}}\right)$, where $\log(\cdot)$ is the logarithm of base 10 and RMS is the root mean square.

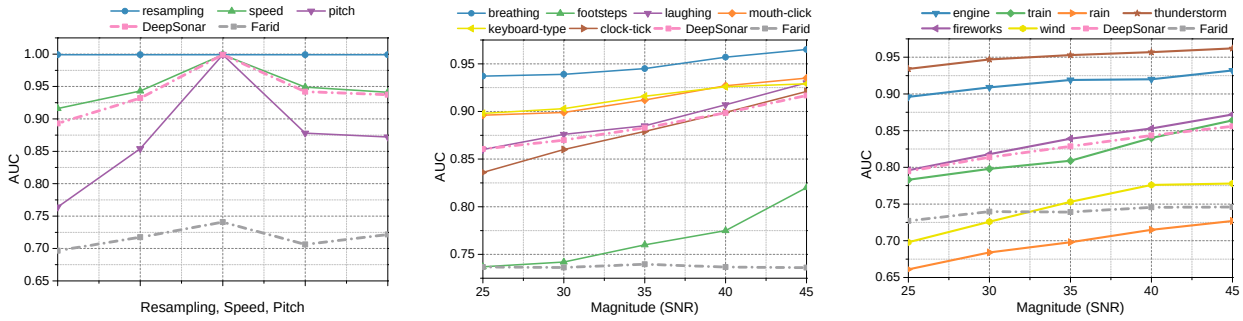
By adding noises to voice data, we first need to obtain the RMS of the noises and voices, respectively. Then, we modify the noise by multiplying each element with a constant to change the RMS, thus the desired SNR is achieved. In voice conversion, various voice manipulations are implemented with the APIs provided by libsora [40]. Figure 5 presents a spectral centroid visualization of the two manipulation attacks, the left is voice conversion and the right is additive real-world noises. The two manipulations all have obvious modifications to the signals, which poses challenges to detectors.

Results on voice conversions. Figure 6(a) shows the experimental results of DeepSonar in tackling three typical voice conversions. We could observe that DeepSonar is robust against resampling including upsampling and downsampling without any performance affected. The average performance is decreased less than 5% and 15% in stretching the voices and shifting pitches, respectively. Compared to the other two conversions (resampling and speed), DeepSonar seems to be a little susceptible to pitch shifting. The main reason is that voices with pitch-shifting have been broken and can hardly listen to the words in voices when the *n_steps* for changing the pitch of voices is larger than 2. The settings for the three voice conversions are presented as follows.

In voice conversion, resampling indicates a time series of voice that is resampled from the original sample rate to the target sample rate, including upsampling and downsampling. Here, the target sample rate is set with an offset (*e.g.*, $-400, 200, 0, 200, 400$) to the original sample rate, where offset 0 servers as a baseline without resampling. Speed represents time-stretch an audio series by a fixed rate. The fixed-rate is set to $0.5, 0.8, 1.0, 1.2, 1.4$, where 1.0 serves as a baseline. Pitch means we shift the pitch of a waveform by n_steps semitones. Here, the n_step is set to $-4, -2, 0, 2, 4$, where n_step 0 serves as a baseline that no pitch is shifted.

Results on indoor-noises. In additive real-world noises, voices are added with representative indoors and outdoors environmental noises. We use SNR to measure the magnitudes of added-noises. In Figure 6(b), DeepSonar performs well on the five indoor noises and the average performance decreased less than 10% at the total five different magnitudes. However, the average performance is decreased by nearly 20% at the five magnitudes when adding footstep noises. We listened to the added-footstep voices which have obviously mixed sizzle noises caused by the friction with floors. Figure 5 also visualizes the differences between original voices and added-footsteps voices.

Results on outdoor-noises. In Figure 6(c), outdoor environmental noises can be roughly classified into three different categories based on the performance of DeepSonar. Engine and thunderstorm environmental noises are the first categories, where the average performance of DeepSonar decreased less than 7% at the five different magnitudes. Fireworks and trains are the second categories, where the average performance of DeepSonar decreased less than 18% at the five different magnitudes. Wind and rain are the third categories, where the average performance of DeepSonar decreased by nearly 25% at the five different magnitudes. We find that environmental noises wind and rain also mixed with other



(a) Voice Conversions (b) Additive Real-world Noises (Indoors) (c) Additive Real-world Noises (Outdoors)

Figure 6: Robustness evaluation of DeepSonar against manipulation attacks at various magnitudes. In voice conversions (a), values in x-axis for resampling are {-400, -200, 0, 200, 400}, for speed are {0.5, 0.8, 1.0, 1.2, 1.4}, for pitch are {-4, -2, 0, 2, 4}. The dotted lines in the three subfigures represent an average performance of our approach DeepSonar and the baseline over different voice manipulations at various magnitudes. Large SNR **means less noises added.**

voices like raindrops on the ground which is much noisy than other types of real-world environmental noises.

According to the experimental results in Figure 6, DeepSonar is also robust against voice conversions except voices are seriously damaged like shifting pitch with a big step. Additionally, DeepSonar performs well when the additive real-world noises are single voice without any mixture with other types of noises. In tackling mixed noises like wind, DeepSonar also holds a high detection performance at the magnitude measured by SNR larger than 35.

Compared with baseline. The dotted lines in the three subfigures of Figure 6 show the comparison results with the baseline by using bispectral artifacts to discern fake voices. To compare the performance of robustness with baseline, we use the average results of the two approaches over different types of voice manipulations at various magnitudes. For example, in Figure 6(b), each point in the dotted line is an average AUC score of the six additive indoor noises at the same magnitude. In Figure 6, the dotted line of DeepSonar is above the baseline, indicating that DeepSonar significantly outperforms the baseline in the two manipulation attacks.

5.3 Discussion

DeepSonar achieves competitive results in terms of both effectiveness, and robustness against two manipulation attacks. However, DeepSonar also exhibits some limitations. First, in adversarial environments, adversaries could add an additional loss function by modeling the neuron behaviors to generate adversarial voices and evade detection. However, most learning-based approaches suffer this adversarial noise attack and an obvious trade-off between generating adversarial voices and evading detection exists. Secondly, real-world noises with a mixture of other types of noises at a high magnitude could decrease the performance of DeepSonar to some extent. Voice denoising will be a potential strategy for high-intensity mixed noises, which would be our future work to remove additional environmental noises. Especially, the voice denoising component is effective without obtaining any prior knowledge of the noises in the complex environments.

6 CONCLUSIONS

In this paper, we proposed DeepSonar that discerns AI-synthesized fake voices by monitoring the learnt neuron behaviors from voice synthesis system. Overall, our work presents a new insight for detecting AI aided multimedia fakes by monitoring neuron behaviors, which aims to build an effective and robust detector. Experiments on the three datasets demonstrate its effectiveness and robustness, with potential in the real-world noisy environment. In fighting against AI-synthesized voices in the wild, robustness should be considered as a priority in designing a detector, since various manipulations on voices can be easily camouflaged as regular operations, while manipulation on images is limited and easy to be spotted. Furthermore, the inconsistency of audio and visual in video DeepFakes is an important clue for DeepFake forensics, thus how to combine recent advances in fake still image and fake voice detection to spot the inconsistency is an important topic for future research. Our neuron behaviors based technique may be a promising idea. Producing and fighting fakes in the AI era is like a mouse and cat game. More powerful techniques should be continuously developed for fighting AI aided fakes as new techniques for producing various fakes will emerge inadvertently. Our future work would continuously investigate how the proposed DeepSonar method can be extended to or work in tandem with various detectors [3, 13, 14, 48, 57] on other modalities of the 'fakes' such as AI-generated / forged images, and DeepFake videos, *etc.*

ACKNOWLEDGMENTS

This research was supported in part by Singapore National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001, National Satellite of Excellence in Trustworthy Software System No. NRF2018NCR-NSOE003-0001, NRF Investigatorship No. NRFI06-2020-0022. It was also supported by JSPS KAKENHI Grant No. 20H04168, 19K24348, 19H04086, and JST-Mirai Program Grant No. JPMJMI18BB, Japan. We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) to our research.

REFERENCES

[1] Ehab A AlBadawy, Siwei Lyu, and Hany Farid. 2019. Detecting AI-Synthesized Speech Using Bispectral Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 104–109.
[2] Christopher M Bishop. 1994. Mixture density networks. (1994).
[3] P. Buchana, I. Cazan, M. Diaz-Granados, F. Juefei-Xu, and M.Savvides. 2016. Simultaneous Forgery Identification and Localization in Paintings Using Advanced Correlation Filters. In *ICIP*.
[4] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. 2019. Activation Atlas. *Distill* (2019). https://doi.org/10.23915/distill.00015

https://distill.pub/2019/activation-atlas.
[5] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*. 5933–5942.
[6] Alexander Comerford. 2019. Detecting audio deep fakes with bispectral analysis. https://github.com/cmrfrd/DetectingDeepFakes_BlackHat2019.
[7] Amazon Corporation. 2020. Amazon AWS Polly. https://aws.amazon.com/polly.
[8] Baidu Corporation. 2020. Baidu Text-to-Speech System. https://cloud.baidu.com/product/speech/tts.
[9] Google Corporation. 2020. Google Cloud TTS. https://cloud.google.com/text-to-speech.
[10] DeepMind. 2020. WaveNet: A generative model for raw audio. https://deepmind.com/blog/article/wavenet-generative-model-raw-audio.
[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[12] Yu Gu and Yongguo Kang. 2018. Multi-task WaveNet: A Multi-task Generative Model for Statistical Parametric Speech Synthesis without Fundamental Frequency Conditions. *Proc. Interspeech 2018* (2018), 2007–2011.
[13] Yihao Huang, Felix Juefei-Xu, Run Wang, Qing Guo, Lei Ma, Xiaofei Xie, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. 2020. FakePolisher: Making DeepFakes More Detection-Evasive by Shallow Reconstruction. *ACM International Conference on Multimedia (ACM MM)* (2020).
[14] Yihao Huang, Felix Juefei-Xu, Run Wang, Qing Guo, Xiaofei Xie, Lei Ma, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. 2020. FakeLocator: Robust Localization of GAN-Based Face Manipulations. *arXiv preprint arXiv:2001.09598* (2020).
[15] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. 2020. DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection. In *CVPR*.
[16] Anne Johnson and Emily Grumbling. 2019. *Implications of Artificial Intelligence for Cybersecurity: Proceedings of a Workshop*. The National Academies Press, Washington, DC. https://doi.org/10.17226/25488
[17] The Wall Street Journal. 2019. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case. https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402.
[18] Shiyin Kang, Xiaojun Qian, and Helen Meng. 2013. Multi-distribution deep belief network for speech synthesis. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8012–8016.
[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958* (2019).
[20] Kazuhiro Kobayashi and Tomoki Toda. 2018. sprocket: Open-Source Voice Conversion Software.. In *Odyssey*. 203–210.
[21] Bruce E Koenig and Douglas S Lacey. 2012. Forensic authenticity analyses of the header data in re-encoded WMA files from small Olympus audio recorders. *Journal of the Audio Engineering Society* 60, 4 (2012), 255–265.
[22] Runnan Li, Zhiyong Wu, Xunying Liu, Helen Meng, and Lianhong Cai. 2017. Multi-task learning of structured output layer bidirectional LSTMs for speech synthesis. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5510–5514.
[23] Yuezun Li and Siwei Lyu. 2018. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656* (2018).
[24] Zhen-Hua Ling, Li Deng, and Dong Yu. 2013. Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 7825–7829.
[25] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling. 2018. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. *arXiv preprint arXiv:1804.04262* (2018).
[26] Jorge C Lucero, Jean Schoentgen, and Mara Behlau. 2013. Physics-based synthesis of disordered voices.. In *Interspeech*. 587–591.
[27] Lei Ma, Felix Juefei-Xu, Minhui Xue, Bo Li, Li Li, Yang Liu, and Jianjun Zhao. 2019. DeepCT: Tomographic Combinatorial Testing for Deep Learning Systems. *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)* (2019).
[28] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. 2018. DeepGauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. ACM, 120–131.
[29] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. 2018. DeepMutation: Mutation testing of deep learning systems. In *ISSRE*.
[30] Shiqing Ma and Yingqi Liu. 2019. NIC: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019)*.
[31] Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, and Ananth Grama. 2018. MODE: automated neural network model debugging via state differential analysis and input selection. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 175–186.
[32] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5188–5196.
[33] Baidu Microsoft. 2020. Baidu TTS. https://www.home-assistant.io/components/tts.baidu.
[34] Yisroel Mirsky and Wenke Lee. 2020. The Creation and Detection of Deepfakes: A Survey. *arXiv preprint arXiv:2004.11138* (2020).
[35] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. 2019. TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing. In *International Conference on Machine Learning*. 4901–4911.
[36] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature Visualization. *Distill* (2017). https://doi.org/10.23915/distill.00007 https://distill.pub/2017/feature-visualization.
[37] Online. 2020. A public lecture record Chinese dataset. http://speech.ee.ntu.edu.tw/~yangchiyi/lecture_tts_data.tgz.
[38] Online. 2020. Ancient Chinese Poetry. https://github.com/chinese-poetry/chinese-poetry.
[39] Online. 2020. Audio Samples: Comparison among Different Synthesis Methods. http://www.ai1000.org/samples/index.html.
[40] Online. 2020. libsora: audio and music processing in Python. https://librosa.github.io.
[41] Online. 2020. Online Fake Images. https://thispersondoesnotexist.com.
[42] Online. 2020. Online Fake Voices. https://ttsdemo.com.
[43] Online. 2020. The Audio Processing Techniques Lab at York. http://bil.eecs.yorku.ca/aptly-lab.
[44] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
[45] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).
[46] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. Deepxplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 1–18.
[47] Karol J. Piczak. [n.d.]. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia* (2015-10-13). ACM Press, 1015–1018. https://doi.org/10.1145/2733373.2806390
[48] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. 2020. DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms. *ACM International Conference on Multimedia (ACM MM)* (2020).
[49] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4779–4783.
[50] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.
[51] Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems*. 7717–7728.
[52] Paul Taylor. 2009. *Text-to-speech synthesis*. Cambridge university press.
[53] Tomoki Toda, Ling-Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. 2016. The Voice Conversion Challenge 2016.. In *Interspeech*. 1632–1636.
[54] Massimiliano Todisco, Xin Wang, Md Sahidullah, He'ctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. 2019. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *Proc. of Interspeech 2019*.
[55] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. 2020. DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. *arXiv preprint arXiv:2001.00179* (2020).
[56] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*. 4790–4798.
[57] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. 2020. FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces. *International Joint Conference on Artificial Intelligence (IJCAI)* (2020).
[58] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards End-to-End Speech Synthesis. *Proc. Interspeech 2017* (2017), 4006–4010.

[59] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman. 2019. Utterance-level Aggregation For Speaker Recognition In The Wild. In *International Conference on Acoustics, Speech, and Signal Processing (Oral)*.

[60] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. 2019. DeepHunter: A Coverage-Guided Fuzz Testing Framework for Deep Neural Networks. In *ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*.

[61] Xin Yang, Yuezun Li, and Siwei Lyu. 2019. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8261–8265.

[62] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems* 30, 9 (2019), 2805–2824.

[63] Mohammed Zakariah, Muhammad Khurram Khan, and Hafiz Malik. 2018. Digital multimedia audio forensics: past, present and future. *Multimedia tools and applications* 77, 1 (2018), 1009–1040.

[64] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda. 2007. The HMM-based speech synthesis system (HTS) version 2.0.. In *SSW*. Citeseer, 294–299.

[65] Hong Zhao and Hafiz Malik. 2013. Audio recording location identification using acoustic environment signature. *IEEE Transactions on Information Forensics and Security* 8, 11 (2013), 1746–1759.