Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7291–7297
Marseille, 20-25 June 2022

Adversarial Speech Generation and Natural Speech Recovery for Speech Content Protection

Sheng Li[1], **Jiyi Li**[2], **Qianying Liu**[3], **Zhuo Gong**[4]
[1]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
[2]University of Yamanashi, Kofu, Japan
[3]Kyoto University, Kyoto, Japan
[4]The University of Tokyo, Tokyo, Japan
Email:sheng.li@nict.go.jp, jyli@yamanashi.ac.jp, ying@nlp.ist.i.kyoto-u.ac.jp, gongzhuo@gavo.t.u-tokyo.ac.jp

Abstract

With the advent of the General Data Protection Regulation (GDPR) and increasing privacy concerns, the sharing of speech data is faced with significant challenges. Protecting the sensitive content of speech is the same important as the voiceprint. This paper proposes an effective speech content protection method by constructing a frame-by-frame adversarial speech generation system. We revisited the adversarial examples generating method in the recent machine learning field and selected the phonetic state sequence of sensitive speech for the adversarial examples generation. We build an adversarial speech collection. Moreover, based on the speech collection, we proposed a neural network-based frame-by-frame mapping method to recover the speech content by converting from the adversarial speech to the human speech. Experiment shows our proposed method can encode and recover any sensitive audio, and our method is easy to be conducted with publicly available resources of speech recognition technology.

Keywords: speech content protection, adversarial example, speech recognition system, deep neural network

1. Introduction

With the advance of voice-based human-computer interaction and the development of intelligent devices, such as the Amazon Echo and Apple's Homepod, speech data have become a new dimension of big data. The collection and sharing of real-world speech data enable innovative services and products, such as Apple's Siri and Google Assistant, and foster studies on intelligent algorithms. However, privacy and security concerns may hinder the collection and sharing of real-world speech data. Therefore, with the advent of the General Data Protection Regulation (GDPR) and increasing privacy concerns, the sharing of speech data is faced with significant challenges (Nautsch and et al., 2019).

Currently, researchers focus on how to protect the speaker's identifiable information, which is represented as *voiceprint* (as analogous to fingerprints), contained in the speech. An attacker may commit *spoofing attacks* (Wu and et al., 2015) to the voice authentication systems or *reputation attacks*, such as fake Obama speech (Suwajanakorn and et al., 2017). Several methods (Justin and et al., 2015; Qian and et al., 2018; Srivastava and et al., 2019; Fang and et al., 2019) for anonymizing speakers' identities have been proposed to address these problems.

However, protecting the sensitive content of speech is the same important as the voiceprint. The sensitive speech content may be found within particular keywords or keyphrases such as named entities (places, dates, locations, organizations, etc.), financial and medical details, dirty words, or the entire speech. Determining what would be considered sensitive information ultimately depends on the use-cases. Speech content privacy refers to the ability to conceal or mask sensitive content information within the speech signal. Currently, there are only limited signal processing-based methods proposed to solve this problem (Masato and Yoshihiro, 2011; Kondo and Sakurai, 2014; Phunruangsakao et al., 2020).

Recent studies in machine learning have discovered that the deep neural network (DNN) models are vulnerable to adversarial examples, which are inputs slightly perturbed in a way that intends to mislead the DNN models into making misclassifications (Szegedy et al., 2013). When the target model parameters are fixed, adversarial examples can be crafted by adding perturbation on the input signal following the gradient descent on the given model. Some studies (Carlini and Wagner, 2018; Yuan et al., 2018; Yakura and Sakuma, 2019; Schönherr et al., 2019; Qin et al., 2019) showed that adversarial audio sequences could fool the DNN models to output any targeted text transcript (e.g., a command) intended by the attacker. However, these studies on the adversarial examples of DNN models focus on hiding a voice command into unnoticeable audio, and the generated sound is not a human voice. In contrast to these existing works, our purpose is to protect the sensitive content in the speech.

Our key idea for implementing this protecting purpose is replacing the corresponding sensitive speech content into the adversarial speech by a speech content protection method and recovering the speech content from converting the adversarial audio to the human voice by a speech content recovery method. In this paper, we propose a novel speech content protection method with existing automatic speech recognition (ASR) systems. We revisited the adversarial examples generating method in the recent machine learning field and used the phonetic state sequence of human voices for the adversarial examples generation. Moreover, we propose a neural network-based frame-by-frame mapping method to recover the speech content by converting from the adversarial audio to the human voice. The experiments are done upon the time-delay neural network (TDNN)-based ASR system with the state-of-the-art waveform feature. We make it possible to encode and recover any sensitive audio.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes our proposed method. Section 4 presents the details of the implementation and experimental evaluations. The conclusions and future works are given in Section 5.

2. Related Work

2.1. Automatic Speech Recognition (ASR)

Automatic speech recognition (ASR) is a technique to convert voice to text transcriptions and is one of the core techniques for man-to-machine and machine-to-machine communications. In recent years, ASR techniques have been extensively used in information retrieval and speech-to-text services, such as speech assistant of Apple Siri, home management service of Amazon Alexa/Echo, intelligent search and service assistant of Google Home, the personal assistant of Microsoft Cortana. In those applications, ASR serves as an efficient and smart interface, and the performance of ASR is essential to the applicability of those services.

In a nutshell, ASR maps a spoken audio sequence to a word sequence. Mel-frequency Cepstrum Coefficient (MFCC) (Muda et al., 2010) is the most widely used feature for ASR because of its ability to extrapolate important features, similar to the human ear. The feature vector is then sent to the model for either training or inferencing and gets the recognized text. Under the statistical framework, the problem is formulated as maximizing the posterior probability of a word sequence when observing an audio sequence. The traditional models are hybrid models such as the Gaussian-mixture model in combination with hidden Markov model (GMM-HMM) (Rabiner, 1988) or deep neural network with hidden Markov model (DNN-HMM) (Dahl et al., 2012). These hybrid models all consist of two independently optimized components: the acoustic and language models.

Modern ASR models follow an end-to-end framework that integrates the two components (e.g., acoustic model and language model) into a single trainable network (Graves et al., 2006; Graves and Jaitly, 2014; Watanabe et al., 2018; Chan et al., 2016; Vaswani et al., 2017). The output words or characters can be treated as labels in these models. However, it relies on the bidirectional long short term memory (BLSTM) recurrent neural network or attention structure which transverses a whole utterance from both directions to estimate frame-wise outputs. It leads to severe time latency and makes the model impractical for adversarial example generation. Another problem with these models is that they can't define where to insert adversarial examples.

For these reasons, we use Time-delay neural network (TDNN) (Waibel et al., 1989; Peddinti et al., 2015) and frame-level tied-triphone-state labels to train the acoustic model with cross-entropy instead. TDNN is quite popular in the industry because it has no recurrent structure and can accelerate the feed-forward and back-propagation. When training the adversarial examples, we need this feature of TDNN. Moreover, we can use frame-level labels to supervise the adversarial example training according to speech content.

2.2. Adversarial Examples to ASR systems

Adversarial examples, the most important technology in this paper, have been extensively studied in the image domain for DNN image classifiers. When the target model parameters are known (e.g., in a white-box setting), adversarial examples can be crafted by one or more steps of perturbation on the input image following the adversarial gradient towards maximizing the classification error of the target model. Adversarial traffic signs with adversarial stickers (Eykholt et al., 2017), and adversarial eye-glass frames (Sharif et al., 2016) have been shown to mislead a DNN traffic sign classifier and facial recognition models, respectively.

Early studies on crafting audio adversarial examples against ASR systems used either a genetic algorithm (Alzantot et al., 2017) or the optimization of a probabilistic loss function (Cisse et al., 2017). These early attempts are untargeted attacks that mislead the model to translate the adversarial audio into incorrect transcripts. The DolphinAttack (Zhang et al., 2017) is a targeted attack that can fool the model into recognizing an inaudible adversarial ultrasound signal to generate a specific transcript that is in the attacker's interest. Voice commands can also be disguised as a form of noise that sounds meaningless to humans (Carlini et al., 2016; Abdullah et al., 2019). Carlini et al. (Carlini and Wagner, 2018) extended the untargeted attacks into targeted attacks against an end-to-end DeepSpeech model (Graves and Jaitly, 2014), by directly perturbing the audio waveform in a white-box setting (Mozilla's implementation). The adversarial audio sequences can fool the DNN model into outputting any targeted text transcript intended by the attacker. Alternatively, CommanderSong (Yuan et al., 2018) attack injects a voice command into a song to mount a targeted attack. Yakura et al. (Yakura and Sakuma, 2019) simulated the transformation of a physical-world recording to construct robust physical audio attacks of two or three words. Two recent studies (Schönherr et al., 2019; Qin et al., 2019) have proposed methods to craft

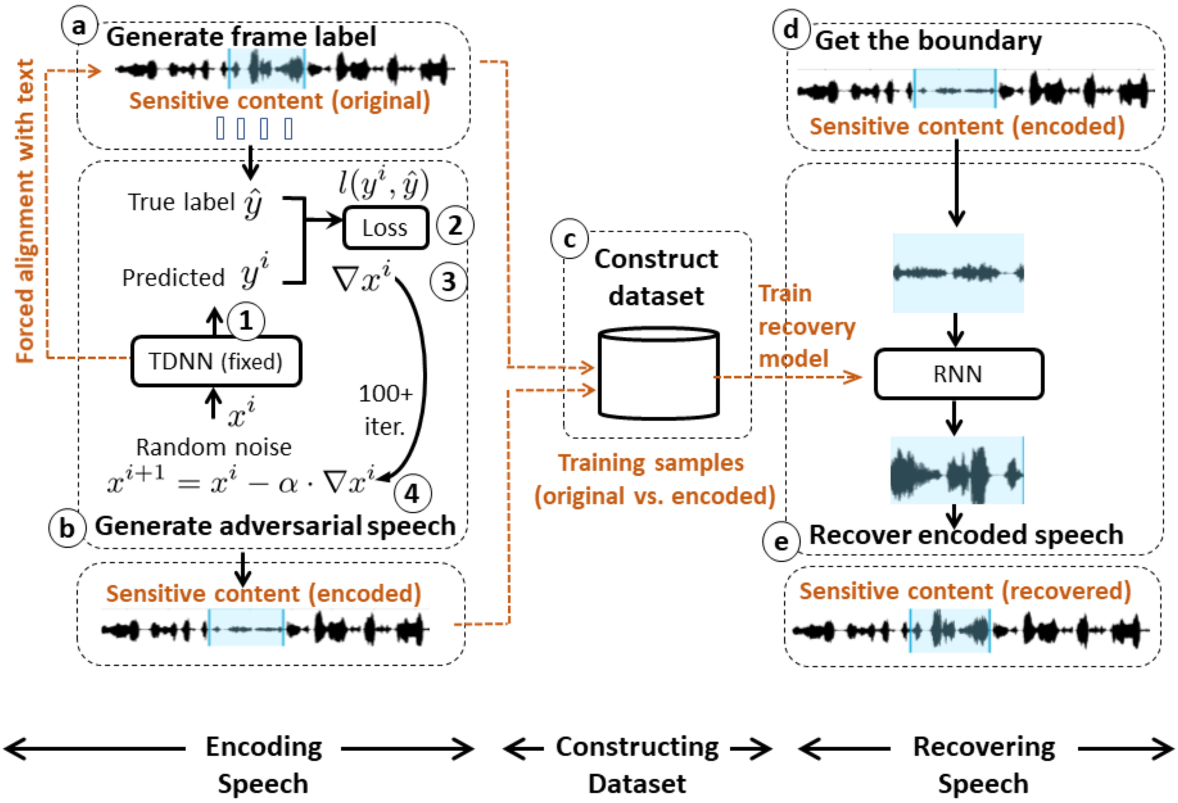

Figure 1: Framework of the proposed method.

imperceptible audio adversarial examples according to the psychoacoustic characteristics of the human auditory system.

3. Proposed Approach

The motivation of our work is protecting the sensitive speech content. Our key idea for implementing this purpose is replacing the corresponding sensitive speech content into the adversarial speech by a speech content protection method, and recovering the speech content from converting the adversarial audio to the human voice by a speech content recovery method. In this paper.

The proposed method is described in Figure 1. To encode and recover the speech with sensitive contents, the proposed method has three steps "encoding speech", "constructing database", and "recovering speech".

For encoding speech, we use the adversarial examples generating method. We generate the state sequence from human voices (**generate frame label** as shown in Figure 1.(a)) and iteratively train the adversarial examples supervised with state label for every frame (**generate adversarial speech** as shown in Figure 1.(b)). We use a TDNN-based acoustic model for ASR and denote it as $f(\cdot)$ (as represented by a TDNN), which maps audio input to a transcription. Given an audio input x (random noise) and an existing TDNN-based acoustic model $f(\cdot)$, we propose constructing a waveform $\hat{x}$ that will make the ASR system output target transcription $\hat{y}$.

We generate adversarial speech for every sentence in public speech dataset, such as Librispeech data (**constructing database** as shown in Figure 1 (c)). So that we can train a DNN model to convert the adversarial speech to natural speech.

For recovering speech, we first get the boundary of the sensitive speech (**get the boundary** as shown in Figure 1.(e)), and then use a frame-by-frame frequency mapping step (**recover encoded speech** as shown in Figure 1.(d)) to convert the encoded speech to human voice.

The steps are detailed in the following subsections.

3.1. Frame Label Generation

We obtain the word-level forced-alignment with time stamps (Young et al., 2009) on the text and speech using a pretrained TDNN acoustic model. So that, we can estimate the boundary of the sensitive contents. Then, we generate the natural state sequence according to human voices as shown in Figure 1.(a). We use a frame-level (tied-triphone-)state sequence as the target label $\hat{y}$=[s_1, s_2, s_3, ...s_m] (m is the frame number of the sentence) as shown in Figure 2. The duration of each frame is 10 msec. The s_t ($1 \leq t \leq m$) is the (tied-triphone-)state id.

We obtain the state-level forced-alignment (Young et al., 2009) on the text and speech using a pretrained

TDNN acoustic model as shown in Figure 1 (a). We also obtain the lengths of the sensitive content and use random noise with a length equal to that as the seed audio x.

3.2. Adversarial Speech Generation

We iteratively train the adversarial examples supervised with a state/frame label for every frame as shown in Figure 1.(b). In contrast to conventional model training, the parameters are kept unchanged, and only input waveform x is updated in our method. We first compute the actual network output y^i,

$$y^i = f(x^i). \tag{1}$$

The difference between the actual network output y^i and label $\hat{y}$ is measured with a loss function $l(y^i, \hat{y})$. Here, i is the current iteration step. We use cross-entropy as the loss function.

$$l(y^i, \hat{y}) = -\sum y^i \log \hat{y}. \tag{2}$$

The gradient of the loss is calculated and back-propagated to the input audio x_i of the DNN.

$$\nabla x^i = \frac{\partial l(y^i, \hat{y})}{\partial x^i}. \tag{3}$$

The input audio x is updated according to the gradient and learning rate α.

$$x^{i+1} = x^i - \alpha \cdot \nabla x^i. \tag{4}$$

We repeat these steps until a fixed number of iterations n (≥ 100).

3.3. Dataset Construction and Speech Recovery

We train an recurrent neural network (RNN)-based model to convert the encoded speech into corresponding human speech by frame-by-frame frequency mapping as shown in Figure 1.(c) and (e). The adversarial speech x^n ($n \geq 100$) obtained in the last step differs from the human-voice waveform. We generate and collect the adversarial speech for every sentence of the given natural speech dataset. Then, based on the collected adversarial speech and the corresponding natural speech, we train an RNN-based conversion model to map the generated adversarial speech x^n and their original human voice $\hat{x}$. The mean absolute error (MAE) is used as a loss function for training regression models. MAE is the sum of absolute differences between the target and predicted variables. We used it to measure the average magnitude of errors in a set of predictions during the training process. The detailed settings are described in Subsection 4.3.

4. Experiments

4.1. Implementation of ASR System

As we mentioned in Subsection 2.1, previous methods (Qin et al., 2019) and (Carlini and Wagner, 2018) use the connectionist temporal classification (CTC) framework (Graves et al., 2006), which is an effective end-to-end framework (Graves and Jaitly, 2014) for speech recognition. However, it relies on the bidirectional long short term memory (BiLSTM) recurrent neural network, which leads to serious time latency. Another problem of the CTC model is that it cannot determine precise time stamps at which the waveform should be generated.

We use a TDNN-based acoustic model with frame-level tied-triphone-state labels and train the model with cross-entropy instead. The TDNN has no recurrent structure, which accelerates the feed-forward and back-propagation processes for adversarial training. Moreover, we can use frame-level labels to supervise the adversarial example training by indicating the precise time stamps at which the waveform should be generated.

We trained the acoustic model using 100 hours of Librispeech data (train-clean-100) (Panayotov et al., 2015). We first trained a GMM-HMM model to derive the state alignment as the training labels. Then, we trained a TDNN with p-norm nonlinearity (p=2) and four layers, each comprising 2,048 hidden nodes following (Peddinti et al., 2015). The output layer had about 3,456 nodes that corresponded to the states of the GMM-HMM model. Instead of using MFCC features, we used 256-dimension raw waveform features (16,000 kHz, 16 bits, mono-channel). All these features were mean normalized (CMN) per speaker. We added a set of layers to perform the log-spectrum feature extraction as shown in Figure 2. All these processes were implemented using the Kaldi toolkit (Povey et al., 2011).

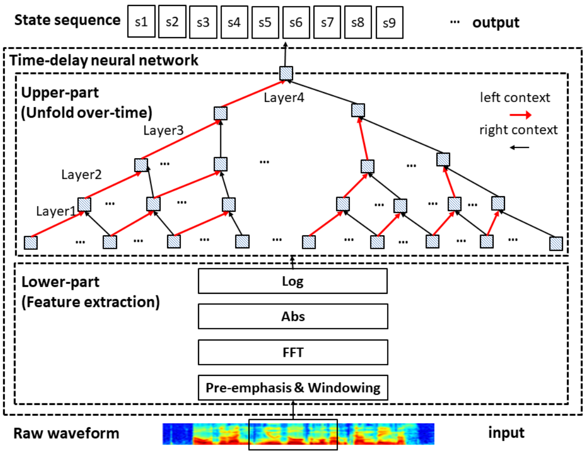

Figure 2: Network of the acoustic model in the ASR system.

The word error rates (WER%) of our TDNN model are around 14.8% (Clean-Dev) to 14.2% (Clean-Test) using the existing trigram word language model (tgsmall) and 200K word vocabulary from the Librispeech opensource resources[1].

4.2. Dataset Construction

We generate adversarial speech for every sentence in 100 hours of Librispeech data (train-clean-100), and select the adversarial speech with the original human voices in total 500 sentence pairs (2 hours for two gender individually) to train the model.[2]

4.3. Implementation of Recovery Model

We train 3-layer RNNs for frame-level mapping from the generated adversarial speech to the human speech of selected data. All the clean speech and generated adversarial speech were down-sampled to 16kHz. The frame length was 32 msec (512 samples), and the frameshift was 16 msec (256 samples). We extracted the 129-dimension log-power spectral feature for training. We also spliced the frames with a context window of seven frames (three left, one center, three right). The mean absolute error (MAE) is used as a loss function for training. The other settings are the default settings[3]. We train gender-dependent models for males and females individually.

4.4. Evaluation

Figure 3 is an empirical evaluation of our proposed encoding and recovery method. The areas in the boxes in Figure 3 are the spectra related to our proposed method. The result of the experiment shows that our proposed method can encode human-speech spectral structure in Figure 3 (top) to adversarial speech in Figure 3 (middle) with significant changes. It can recover the human-voice spectral structure in Figure 3 (bottom).

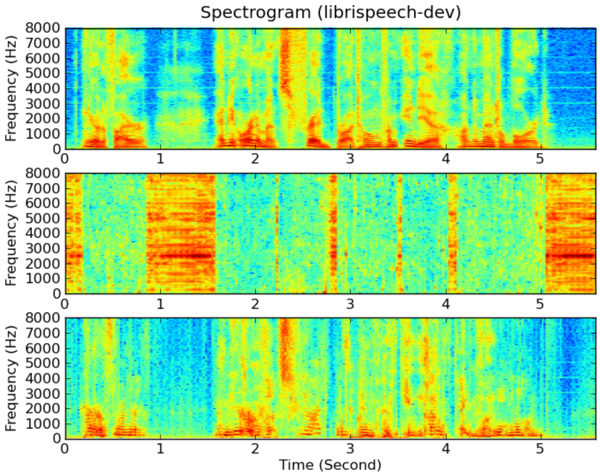

Figure 3: Spectra of a natural human speech (top), the encoded human speech (adversarial speech) (middle), and the recovered human speech (bottom).

For more detailed evaluation, we compare the proposed method with the voices recovered by End-to-End speech synthesis toolkit (Watanabe et al., 2018) on default settings[4]. The acoustic model is trained with train-clean-100 based on Tacotron-2 (Wang and et. al, 2017), which is a sequence-to-sequence model with an attention mechanism and X-vector-based speaker-embeddings. The vocoder uses the Griffin-Lim algorithm (Griffin and Lim, 1984) to speed up the work process. We name it as Tacotron2+Griffin-Lim.
For objective evaluation, we compute the mean squared errors (MSE) of the spectrograms between the original human speech and the synthesized speech using the natural state/frame label from forced-alignment (natural seq.).

Table 1: MSE of two different methods.

	Proposed (natural seq.)	Tacotron2 +Griffin-Lim
Male	0.18	0.75
Female	1.04	0.95

For subjective evaluation, we invite 11 listeners to evaluate the recovered voices. The mean opinion score (MOS) is used to evaluate the generated voices. The raters score the naturalness and intelligibility from 1 to 5. “1” means the poorest results to understand, and “5” is the best result, with 0.5 points increments for every level. We recovered human-voice waveforms using the natural state/frame label (natural seq.).

Table 2: MOS of different methods (95% confidence interval).

	Proposed (natural seq.)	Tacotron2 +Griffin-Lim
Male	3.00 ± 0.85	3.18 ± 0.87
Female	3.47 ± 0.81	4.09 ± 0.52

As shown in Table 1 and 2, the results are quite different between male and female speeches. The imbalance of female training data (less than 20% in total) resulted in larger MSE and lower MOS values. The MOS results of the proposed method for males are comparative to Tacotron2+Griffin-Lim.

4.5. Further Discussions

Experiments show that our proposed method can encode and recover any sensitive audio. The advantage is it is very easy to construct with publicly available resources of speech recognition technology.
We also noticed that training the recovery network is most critical. The model training is sensitive to the accuracy of the adversarial speech and data size. To generate suitable adversarial, we need an acoustic model of the ASR system with high recognition accuracy. The

[1] http://www.openslr.org/11/

[2] The database and the listening samples will be released on the project page of our lab during conference.

[3] https://github.com/yongxuUSTC/sednn

[4] https://github.com/espnet (2018 version)

data should consist of a single person's voice. In the future, we will exploit additional massive single-speaker datasets, such as LJSpeech[5].

Our method is designed for low-latency scenarios. The raw waveform crafting stage mostly determines the speed. In our experiment, we only use a single CPU on the workstation. The time required to run 100 iterations for generating a 10-second-long raw waveform is around 30 seconds. It can be shortened to no more than 2 seconds by splitting the speech into 16 segments to run in parallel. It can be further accelerated by using GPU in batch-mode and a small number of iterations.

5. Conclusion and Future Plan

In this paper, we revisit the adversarial examples generating method in the recent machine learning field upon a time-delay neural network (TDNN)-based speech recognition system and used the phonetic state sequence of human speech for the adversarial speech generation. The adversarial speech can be used to protect the sensitive speech content in a frame-by-frame manner precisely. Moreover, we also build adversarial-human parallel speech corpus to train neural networks and recover the protected speech content to human speech. The experiment shows our proposed method effectively encodes and recovers any sensitive audio using publicly available speech recognition resources. In the future, we will introduce state-of-the-art speech synthesis technology to enhance the recovery model.

6. Acknowledgements

We thank Hao Shi of Kyoto University and Yaowei Han of Google, Tokyo, for implementing and testing the recovering model. The work is partially supported by JSPS KAKENHI Grant No. 21K17837, JST SPRING Grant No. JPMJSP2110 and NICT tenure-track startup funding.

7. Bibliographical References

Abdullah, H., Garcia, W., Peeters, C., Traynor, P., Butler, K. R. B., and Wilson, J. (2019). Practical hidden voice attacks against speech and speaker recognition systems. *NDSS*.

Alzantot, M., Balaji, B., and Srivastava, M. (2017). Did you hear that? adversarial examples against automatic speech recognition. *NIPS 2017 Machine Deception workshop*.

Carlini, N. and Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. abs/1801.01944.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. (2016). Hidden voice commands. In *Proc. USENIX*, pages 513–530.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. IEEE-ICASSP*.

Cisse, M. M., Adi, Y., Neverova, N., and Keshet, J. (2017). Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Proc. NIPS*, pages 6977–6987.

Dahl, G., Yu, D., Deng, L., and Acero, A. (2012). Context dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Trans. ASLP*, 20(1):30–42.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2017). Robust physical-world attacks on machine learning models. *arXiv*.

Fang, F. and et al. (2019). Speaker anonymization using X-vector and neural waveform models. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 155–160.

Graves, A. and Jaitly, N. (2014). Towards End-to-End speech recognition with recurrent neural networks. In *Proc. ICML*.

Graves, A., Fernandez, S., Gomez, F., and Shmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*.

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Trans. ASSP*, 32(2):236–243.

Justin, T. and et al. (2015). Speaker de-identification using diphone recognition and speech synthesis. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, volume 4, pages 1–7. IEEE.

Kondo, K. and Sakurai, H. (2014). Gender-dependent babble maskers created from multi-speaker speech for speech privacy protection. In *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 251–254.

Masato, A. and Yoshihiro, I. (2011). Privacy protection for speech based on concepts of auditory scene analysis.

Muda, L., Begam, M., and Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*.

Nautsch, A. and et al. (2019). The GDPR & speech data: Reflections of legal and technology communities, first steps towards a common understanding.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *Proc. IEEE-ICASSP*.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. INTERSPEECH*.

Phunruangsakao, C., Kraikhun, P., Duangpummet, S.,

[5] https://keithito.com/LJ-Speech-Dataset/

Karnjana, J., Unoki, M., and Kongprawechnon, W. (2020). Speech privacy protection based on controlling estimated speech transmission index. In *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 628–631.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *Proc. IEEE-ASRU*.

Qian, J. and et al. (2018). Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 82–94. ACM.

Qin, Y., Carlini, N., Goodfellow, I., Cottrell, G., and Raffel, C. (2019). Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proc. ICML*, volume 97, pages 5231–5240, 09–15 Jun.

Rabiner, L. R. (1988). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Schönherr, L., Kohls, K., Zeiler, S., Holz, T., and Kolossa, D. (2019). Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *NDSS*.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM Conference on Computer and Communications Security (CCS)*, pages 1528–1540.

Srivastava, B. and et al. (2019). Evaluating voice conversion-based privacy protection against informed attackers. *arXiv preprint arXiv:1911.03934*.

Suwajanakorn, S. and et al. (2017). Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics*, 36(4):95.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *CoRR abs/1706.03762*.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE/ACM Trans. ASLP*, 37(3):328–339.

Wang, Y. and et. al. (2017). Tacotron: Towards end-to-end speech synthesis. In *Proc. INTERSPEECH*.

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N. E. Y., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., and Ochiai, T. (2018). Espnet: End-to-end speech processing toolkit. In *Proc. INTERSPEECH*.

Wu, Z. and et al. (2015). Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153.

Yakura, H. and Sakuma, J. (2019). Robust audio adversarial example for a physical attack. *Proc. IJCAI*.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2009). The HTK book version 3.4.1. In *Tutorial Books*.

Yuan, X., Chen, Y., Zhao, Y., Long, Y., Liu, X., Chen, K., Zhang, S., Huang, H., Wang, X., and Gunter, C. A. (2018). Commandersong: A systematic approach for practical adversarial voice recognition. In *Proc. USENIX*, pages 49–64.

Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. (2017). Dolphinattack: Inaudible voice commands. In *ACM Conference on Computer and Communications Security (CCS)*, pages 103–117. ACM.