Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

INOR—An Intelligent noise reduction method to defend against adversarial audio examples

Qingli Guo[a,b,*], Jing Ye[a,b], Yiran Chen[c], Yu Hu[a,b], Yazhu Lan[a], Guohe Zhang[d], Xiaowei Li[a,b]

[a] State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, PR China
[b] University of Chinese Academy of Sciences, Beijing 100049, PR China
[c] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
[d] School of Microelectronics, Xi'an Jiaotong University, Xi'an, Shanxi 710049, PR China

ARTICLE INFO

Article history:
Received 24 August 2019
Revised 13 December 2019
Accepted 29 February 2020
Available online xxx

Communicated by Dr. Chenchen Liu

Keywords:
Adversarial audio examples
Defense against adversarial audio examples
INOR

ABSTRACT

Recently, Automatic Speech Recognition(ASR) systems are seriously threatened by adversarial audio examples. The defense against adversarial audio examples has become an urgent issue. Different from adversarial image examples whose target is limited in the finite categories, the target of adversarial audio examples can be any combination of the words in a language. Adversarial audio examples aim to change the semantic of the audio. The semantic is explicitly represented in transcription distance, which affects the adversarial perturbation. This paper analyzes the relationship between semantic difference and adversarial perturbation. Quantization and local smoothing are calibrated to evaluate their performance. We observe that, for adversarial audio examples with different transcription distance levels, the capability of different denoising strategies varies. Therefore, we first introduce the wavelet filter, which denoises the signal in the transformed domain. Then we explore the defense capability of combined filters. Finally, a new intelligent noise reduction method–INOR is proposed to improve the denoising performance of audios under different levels of transcription distance. Experimental results show that INOR is effective in mitigating the adversarial perturbations for adversarial examples with different transcription distance levels. The average CER and WER is reduced by 33% and 55%.

1. Introduction

In recent years, Deep Neural Networks (DNNs) have achieved impressive performance in various artificial intelligent applications including image classification [1], natural language processing [2], and speech recognition [3]. In some areas, DNNs even achieve near-human accuracy [1], so they are widely adopted in security-sensitive tasks which request high robustness of the model.

However, DNNs are seriously threatened by adversarial attack. Adversarial attack aims to generate adversarial examples by adding imperceptible noises to legitimate samples in order to mislead the model into incorrect prediction. Although adversarial examples cannot be perceived by human beings, they may be misclassified by DNNs and cause catastrophic failure in crucial tasks. Besides, adversarial examples can also be leveraged to cheat adversaries which enhances the security of the system [4–6].

Adversarial attacks can be classified into two main categories: the targeted attack and the untargeted attack. The prediction of the targeted attack is specified by the adversary, while the prediction of the untargeted attack can be any one except the clean one. The targeted attack is often implemented by minimizing the loss of target adversarial prediction, while untargeted attack is often implemented by maximizing the loss of the clean prediction. Compared with untargeted attack, targeted attack is more difficult to implement.

Among the studies of adversarial examples, early works mainly focus on the continuous domain, that is, the domain of images. Adversarial image examples are generated to fool image classification system. Adversarial examples was first introduced in image classification by Szegedy et al. [7], which is a targeted attack. Other targeted attacks in image domain include the CW (Carlini&Wagner) attack [8], the iterative FGSM(Fast Gradient Sign Method) attack [9], the impersonation attack [10], and so on. As for untargeted image attacks, there are the FGSM attack [11], the DeepFool attack [12], the Universal Perturbation [13], and so on.

* Corresponding author at: State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, PR China.
E-mail addresses: guoqingli@ict.ac.cn (Q. Guo), yejing@ict.ac.cn (J. Ye), yiran.chen@duke.edu (Y. Chen), huyu@ict.ac.cn (Y. Hu), yulan@ict.ac.cn (Y. Lan), zhangguohe@xjtu.edu.cn (G. Zhang), lxw@ict.ac.cn (X. Li).

https://doi.org/10.1016/j.neucom.2020.02.110
0925-2312/

However, where there is the attack, there is the defense. Various countermeasures have been proposed to defend against adversarial image examples. Adversarial training [9,11,14] and network distillation [15] aim to improve the robustness of the neural network model in the training stage so that the model can output the correct prediction when an adversarial image example comes. Adversarial detecting [16–19] tries to detect adversarial image examples. Most adversarial detection strategies aim for obtaining a model which can classify the input image as a clean one or an adversarial one. Adversarial training, network distillation, and adversarial detection all require modifying the neural network model or training new sub-models. Another countermeasure is to transform the input image, such as JPEG compression [20] and smoothing filters [5,21]. They modify the input image to mitigate perturbations, so the classification output will not be affected by adversarial image examples.

Not until recently, researchers began to pay attention to adversarial audio examples, whose domain is discrete. Adversarial audio examples are generated to fool the Automatic Speech Recognition (ASR) system, which converts a spoken utterance to a text transcription. Microsoft Cortana [22], Apple Siri [23], Google Now [24], Amazon Alexa [25], CMU Sphinx [26], Kaldi [27], and DeepSpeech [28] are all popular ASR systems. The Deepspeech is an end to end system based on the DNN. Similar to adversarial image examples, the works related to adversarial audio examples mainly contain two topics: the adversarial attack and the adversarial defense.

Cisse et al.[29–31] craft untargeted adversarial audio examples, whose transcription is different from the transcription of the clean audio. Targeted adversarial audio examples can be generated using the methods proposed in [29,32–35]. Carlini et al. [32] generates hidden voice commands that are unintelligible to human beings but are intelligible to ASR system. However, hidden voice commands are easy to be recognized by humans because they are noise like and are audible to humans. Exploiting the circuit nonlinearity of microphone, Zhang et al. [33] is able to modulate voice commands that are audible to devices but inaudible to humans. Fortunately, low-pass filters and classification model can be applied to mitigate or detect the command signals because they are ultrasonic. Both [32,33] synthesize specific audios for attacks. Cisse et al. [29] proposes a more powerful attacking method, named Houdini, which can generate targeted adversarial audio examples by modifying existing audios, but Houdini can only generate adversarial examples whose transcriptions are phonetically close to the clean ones. Later Carlini et al propose a more powerful targeted attack in [32] that can target any adversarial transcriptions. The generated adversarial audio examples are hard to be distinguished by humans and are against the state-of-the-art ASR system–DeepSpeech [28]. Qin et al. [35] makes the noise less perceptible by leveraging “Psychoacoustic Hiding”, but their attack is mounted on Lingvo classifier which is based on the Listen, Attend, and Spell model.

The defense against adversarial audio examples aims to detect adversarial audio examples or recover the clean transcription. For the sake of simplicity, the frequently used terms are explained in Table 1. The transcription of the clean audio is denoted as *clean transcription*, while the transcription of the adversarial audio is referred to as *adversarial transcription*. The *original transcription* refers to the transcription of the original audio. The original audio is the audio that has not been denoised, so it can be clean or adversarial. The *denoised transcription* refers to the transcription of the denoised audio. The *transcription distance* is the distance between the clean transcription and the adversarial transcription. The *semantic difference* is the difference between the semantic of the clean transcription and the adversarial transcription.

Table 1
Description of terms.

Terms	Description
Clean Transcription	The transcription of the clean audio
Adversarial Transcription	The transcription of the adversarial audio
Original Transcription	The transcription of the original audio before being denoised
Denoised Transcription	The transcription of the denoised audio
Transcription Distance	The distance between the clean transcription and the adversarial transcription
Semantic Difference	The difference between the semantics of the clean transcription and the adversarial transcription

Various strategies have been proposed to implement the adversarial audio example recovery. Sun et al. [36,37] leverage data augmentation and adversarial training to improve the model robustness. Yang et al. [38] exploits input transformation to mitigate the effect of adversarial noises and recover the clean audio sequence. Input transformation does not require modification to the model, so it can be directly integrated into the ASR system. However, the quality of the transcription drops by a great degree since the recovered audio sequence is still very different from the clean one, and the transcription quality of clean samples is also defected.

For image classification, the output is the category of the input image, and the adversarial target space of the adversarial image example is limited in the finite categories. Differently, the output of the ASR system is text, and the target of the adversarial example can be any combination of the words in a language. The goal of adversarial audio example is to change the semantic, which is represented explicitly by the transcription distance. However, the semantic difference may not consist with the transcription distance, which further affect the adversarial perturbation. For different adversarial audio examples, transcription distance may be either large or small. Existing defense strategies treat these adversarial examples in the same way, but this may be inappropriate since the magnitude of perturbations varies.

Our contributions include:

1. This paper analyzes the relationship between semantic difference and adversarial perturbation. The quantization and local smoothing are calibrated to evaluate their performance. We observe that, for adversarial audio examples with different transcription distance levels, different denoising strategies have different capabilities.
2. Based on the above observation, we introduce the wavelet filter, and explore the defense capability of combined filters.
3. A new intelligent noise reduction method, INOR, is proposed to improve the denoising performance of audios under different level of transcription distance.
4. The experimental results show that, using INOR, the average CER and WER is reduced by 33% and 55% respectively.

2. Background and related work

2.1. Automatic speech recognition and adversarial audio example

ASR system has been integrated into many commercial applications such as Microsoft Cortana [22], Apple Siri [23], and Google Now [24], and brings enormous convenience to human beings. ASR system takes audio signal as input and output the transcription in the form of text. To help research, many open-source ASR systems have been developed. According to their architectures, these ASR systems can be classified into two categories.

Conventional ASR systems consist of an acoustic model and a language model. The former one models the relationship between audio signals and phonetic units, while the later one models the word sequences in the language. Conventional ASR systems require

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

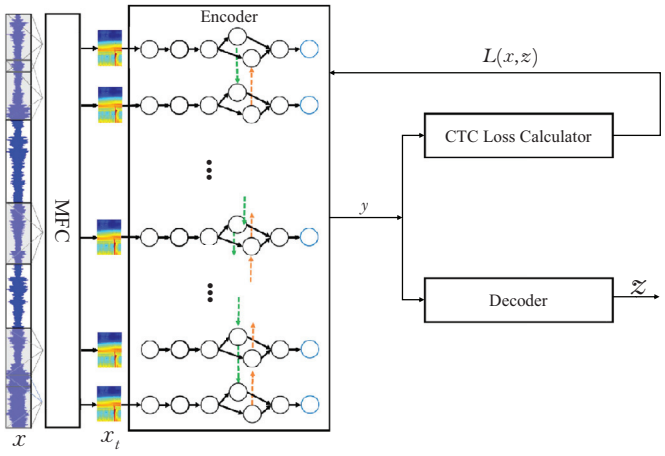


Fig. 1. Overview of DeepSpeech.

laborious process stages since the acoustic model and the language model have to be pretrained separately. Both Kaldi [27] and CMU Sphinx [26] belong to conventional ASR systems. DeepSpeech is a state-of-the-art ASR system which is end-to-end. The input to DeepSpeech is the audio signal represented by a N-dimensional vector–x which is sampled under a specific sampling rate. As shown in Fig. 1, DeepSpeech mainly contains four components: Mel-Frequency Cepstrum (MFC), Encoder, Connectionist Temporal Classification (CTC) Loss Calculator and Decoder. MFC is often used to extract the features of the audio. It is able to approximate the human auditory system's response closely, and achieves better representation of sound. The MFC [39,40] preprocesses the waveform or the audio signal, x, by splitting it into T short frames and extracting the features that are useful for identifying linguistic content. There is an overlap between adjacent frames, and the extracted features is x_t for t'th frame. The encoder takes the extracted feature as input and outputs the probability distribution–y over all the individual characters, and the probability matrix is computed using an RNN model. CTC is a method to train RNN without the requirement for sequence segmentation. CTC loss is calculated in training step based on y with CTC loss calculator. The decoder takes the probability matrix as input and calculates the probabilities of all label sequences, and the phrase with the highest probability is chosen as the final output of the speech recognition system [41]. The whole process performed by DeepSpeech can be denoted as $z = F(x)$. The performance of DeepSpeech is competitive to the conventional ASR systems, but the architecture is much simpler.

However, the security of ASR systems is threatened by adversarial audio examples, which mislead ASR systems into wrong prediction. As shown in Eq. (1), to generate adversarial audio examples, imperceptible perturbation δ, which is also a N-dimensional vector, is added to the clean sample x. δ is carefully designed to change the predicted transcription z. However, since δ is very small, human beings can hardly be aware of it.

$$\begin{aligned} z &= F(x) \\ z' &= F(x+\delta) \\ z' &\neq z \end{aligned} \tag{1}$$

Untargeted attack and targeted attack are two main categories of adversarial audio attack. In untargeted attack, the adversary try to make $z' \neq z$. In targeted attack, z' is not only different from z, but is also assigned by the adversary. Since untargeted attack can be implemented by the targeted attack, targeted attack is more powerful.

Many works [29,32–35] have focused on targeted attack against ASR systems. Both hidden voice commands [32] and DolphinAttack [33] generate new audios instead of modifying existing audios. With the help of MFC and inverse-MFC, hidden voice commands can be generated by removing the features that are helpful for human listeners to comprehend but are not useful in ASR system. Therefore, hidden voice commands are unintelligible to human listeners but can be recognized by ASR systems. DolphinAttack modulates the voice commands on ultrasonic carriers to make the voice commands completely inaudible. Compared with hidden voice commands and DolphinAttack, Houdini [29] is able to construct adversarial audio examples by doing small changes to existing audios with the help of a more effective surrogate loss function. However, Houdini can only generate adversarial audio examples whose transcriptions are phonetically similar to that of the clean samples.

A more powerful targeted attack is proposed by Carlini and Wagner in [8]. They can target any transcription by adding only slight distortion–δ, which is obtained by solving the optimization problem presented in Eq. (2). Here, c is a suitably chosen constant. It is used to minimize the loss towards the target and the distortion towards the original audio. $\ell_{CTC}(x+\delta, z')$ is the loss of the adversarial example towards the target. The optimization problem is solved using gradient-based algorithm.

$$\min \ell(\delta, z') = |\delta|^2 + c \cdot \ell_{CTC}(x+\delta, z') \tag{2}$$

2.2. Defense strategies against adversarial audio examples

Previous works have proposed many strategies to defend adversarial audio examples. Since the adversarial audio examples generated by DolphinAttack is ultrasonic, they can be easily mitigated leveraging low-pass filters. As shown in [33], DolphinAttack can also be detected through classification by support vector machine. Houdini can only generate adversarial audio examples with similar phonemes as the original audio, so its threat is limited. No existing works have paid attention to the defense of Houdini yet.

The attack proposed in [34] is much more dangerous since it can generate adversarial audio examples with any chosen target transcription. Yang et al. [38] leverages the temporal dependency to detect adversarial audio examples. Firstly, the complete audio sequence, x, is input to the ASR system and the transcription–z is got. Secondly, the detector inputs the first $p(1 \leq p \leq 1)$ portion of x to the ASR system and get the partial transcription–z'. Then the distance between the first p portion of z and z' is computed. The distance can be Character Error Rate (CER) and Word Error Rate (WER). WER and CER [42] are common metrics of the efficiency of a speech recognition system, which measures the distance between the denoised transcription and the clean transcription. The Error Rate (ER) is computed according to Eq. (3), where S, D, I is the number of substitutions, the number of deletions, and the number of insertions respectively, which are computed using dynamic string alignment. N is the number of words or characters in the reference transcription, and here is the clean transcription. Finally, the distance is used as the feature for distinguishing adversarial examples from clean samples.

$$ER = \frac{S+D+I}{N} \tag{3}$$

Although the temporal dependency based method may detect adversarial audio examples, it is insufficient for recovering the clean transcription. There are mainly two ways to recover the clean transcription: model robustness improvement and input transformation. Sun et al. [36,37,43] focus on the improvement of the model robustness by data augmentation or adversarial training. However, all of them require modifying the original model. The

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

set0: clean
Example 1: Without the dataset the article is useless
Example 2: We are refugees from the tribal wars and we need money the other figure said
Example 3: The night was warm and I was thirsty

set1: only one of the words are replaced compared with the clean transcription
Example 1: Without the dataset the article is not useless
Example 2: We are refugees from the tribal wars and we do not need money the other figure said
Example 3: The night was not warm and I was thirsty

set2: half of the words are replaced compared with the clean transcription
Example 1: Without notes the music is meaningless
Example 2: They are citizens from the richest country and they do not need food the other people said
Example 3: The day was cold and I was hungry

set3: all of the words are replaced compared with the clean transcription
Example 1: Okay google browse to evil dot com
Example 2: Looking around he sought his sheep and then realized that he was in a new world
Example 3: This is what was written on the emerald tablet said the alchemist when he had finished

Fig. 2. Example transcription.

modification of the model introduces expensive training cost because the training progress is usually time consuming and need a large number of benign examples and corresponding adversarial examples.

Comparatively, input transformation only needs to transform the input audios while leaving the model unchanged. The goal of input transformation is to recover the clean transcription from the adversarial audio examples by corrupting the adversarial perturbation. WER and CER are evaluation metrics to measure the recovery performance. Yang et al. [38] tests the feasibility of input transformation using traditional signal processing methods: *Quantization* and *Local smoothing*. *Quantization* maps the input value from a larger set to a smaller set. In audio quantization, the amplitude of sampled audio signal is rounded into the nearest multiple of an integer–*q*. *Local smoothing* uses a sliding window with a fixed length to determine the value of a sample point. The sliding window contains $k-1$ points before and after a point in an audio, and this point is replaced with the smoothed value such as the median or the average value of the window. However, the performance is still limited. To effectively recover the clean transcription and avoid severe semantic manipulation, this paper proposes an intelligent noise reduction method.

3. Motivation and observation

This section first analyzes the relationship between semantic difference and adversarial perturbation. Based on the analysis, quantization and local smoothing are applied to mitigate adversarial perturbations. The parameters of each strategy are calibrated to evaluate their performance.

3.1. Relationship between semantic difference and adversarial perturbation

In image classification, the target of an adversarial example is a specific class from a limited space of classifications. However, in speech recognition, the target is a text sequence, which can be any combination of characters, so there are numerous possibilities. The goal of adversarial audio examples is to change the semantic, which is represented explicitly in the transcription. With the same semantic difference, the transcription distance varies. The transcription distance may further affect the magnitude of adversarial perturbation, which may then affect the defense difficulty.

Based on the transcription distance, the adversarial audio examples are roughly classified into three categories. Some transcription examples are shown in Fig. 2. The transcriptions in *set*0 are clean. The transcriptions in *set*1, *set*2, and *set*3 are adversarial, but the transcription distances are different. Comparing transcriptions of *set*0 and *set*1, the semantic is reversed, but only one word is changed by adding “not” or deleting “not”. Comparing transcriptions of *set*0 and *set*2, partial words are replaced. Comparing transcriptions of *set*0 and *set*3, all of the words are replaced. Different level of transcription distance may introduce different form of adversarial perturbation, which may enforce different level of defense difficulty. Meanwhile, no matter what the level of transcription distance is, the semantic can be changed seriously.

To observe the differences between the three groups of adversarial examples, their adversarial perturbations are analyzed. For each adversarial audio, the perturbation has a minimum value, a maximum value, a mean value and a median value, which are computed according to Eq. (4)–(7). The minimum value for most perturbations is zero. We select the first 100 instances from the test set of*Mozilla Common Voice* as the clean samples in *set*0. The adversarial audio examples in *set*1, *set*2, and *set*3 are generated according to the adversarial transcriptions.

$$\delta_{min} = min(abs(\delta)) \tag{4}$$

$$\delta_{max} = max(abs(\delta)) \tag{5}$$

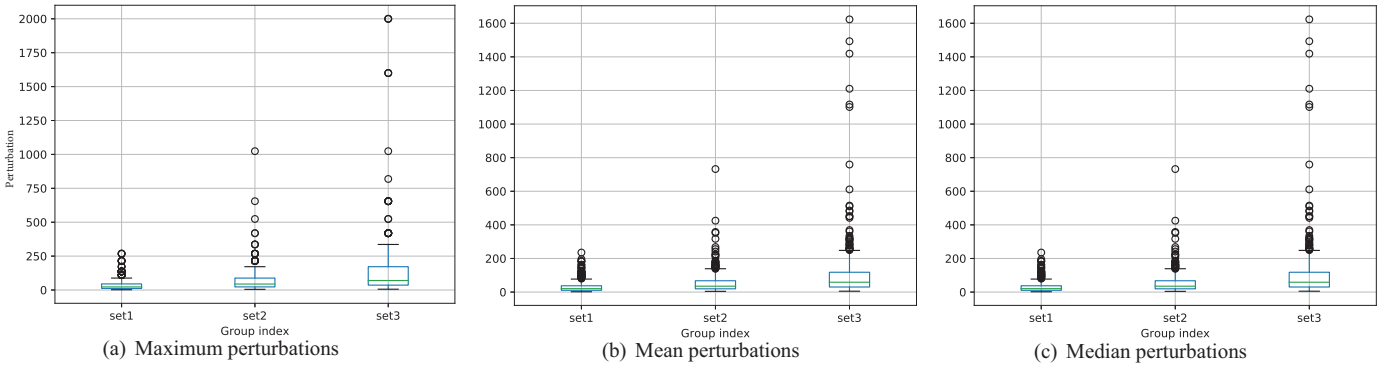
$$\delta_{mean} = mean(abs(\delta)) \tag{6}$$

$$\delta_{median} = median(abs(\delta)) \tag{7}$$

Here, δ is the perturbation; $\delta_{\max}$, δ_{mean}, and δ_{median} are the maximum value, the mean value, and the median value.

The distribution of the maximum, mean, and median perturbations for each set of adversarial examples are shown in Fig. 3. From the figure we can see that, compared with *set*1, most of the maximum perturbations in *set*2 are higher; compared with *set*2, most of the maximum perturbations in *set*3 are higher. The mean perturbations and median perturbations also present the same trend. This

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110



(a) Maximum perturbations (b) Mean perturbations (c) Median perturbations

Fig. 3. Perturbations for each set of adversarial audio examples.

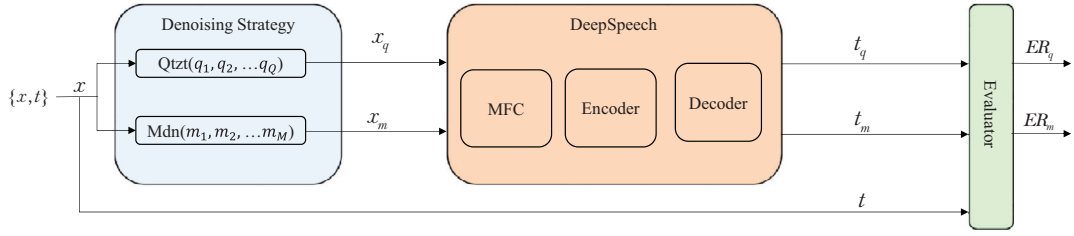


Fig. 4. Parameter Calibration for Traditional Denoising Strategies.

trend implies that, while the transcription distance increases, the adversarial perturbation also increases. The transcription distance is consistent with the adversarial perturbation. However, since the semantic difference is not consistent with transcription distance, it is not consistent with adversarial perturbation. The adversarial examples with different level of perturbations may response differently to the same denoising strategy, which makes it necessary to evaluate the performance of denoising strategies for each set separately. In following sections, denoising strategies are evaluated on the audios in *set*0, *set*1, *set*2, and *set*3.

3.2. Parameter calibration for quantization and local smoothing

Primitive transformation strategies include quantization and local smoothing. Audios with different level of adversarial perturbations may response differently to one denoising strategy. Therefore, to evaluate the performance of the denoising strategies, we apply quantization and local smoothing to the audios in *set*0, *set*1, *set*2, and *set*3 generated in Section 3.1 respectively. It is expected that, the denoising strategy can not only mitigate the perturbation of the adversarial audio examples, but also leave the clean samples not affected. That is, the adversarial transcription is recovered to clean transcription, while the clean transcription is still the clean transcription. Meanwhile, the parameters of the denoising strategies are calibrated to analyze their impacts on performance. With ER as the metric, the denoising strategies are evaluated on various parameters.

The parameters are calibrated according to Fig. 4. The audio x is first denoised using the strategies mentioned above. "Qtzt" is the abbreviation of quantization, and "Mdn" is the abbreviation of median smoothing. $(q_1, q_2, \ldots q_Q)$, $(m_1, m_2, \ldots m_M)$, and $(w_1, w_2, \ldots w_W)$ are the value of the parameters. Q and M are the number of parameter values. The denoised audio are x_q and x_m, which are then sent to DeepSpeech. DeepSpeech helps get the transcriptions of the denoised audio t_q and t_m. The distance between t_q, t_m and the reference transcription t are computed by the Evaluator. The reference transcription is the ground-truth transcription.

Table 2
Performance of quantization.

q	set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	**0.30**	**0.14**	0.18	0.11	0.54	0.43	1.19	0.98
1	0.29	0.13	0.18	0.10	0.55	0.42	1.15	0.96
2	0.31	0.14	0.16	0.08	0.56	0.42	1.13	0.94
3	0.32	0.15	**0.14**	0.06	0.52	0.37	1.09	0.91
4	0.30	0.15	0.16	0.07	0.46	0.32	1.03	0.83
5	0.33	0.16	0.19	0.08	0.45	0.28	0.92	0.70
6	0.40	0.20	0.28	0.13	**0.46**	**0.27**	0.84	0.57
7	0.48	0.27	0.43	0.24	0.52	0.31	0.79	0.52
8	0.61	0.39	0.58	0.36	0.63	0.40	**0.74**	**0.51**
9	0.75	0.56	0.73	0.55	0.74	0.56	0.83	0.62
10	0.90	0.74	0.90	0.74	0.88	0.73	0.91	0.76

3.2.1. Quantization

Quantization rounds the amplitude of sampled audio signal to the nearest multiple of an integer, which is often 2^q, and q is selected by the defender. Quantization maps the value of the input from a larger set to a smaller set. The searching space of q ranges from 1 to 10, with the step of 1. The experimental result for each set of audios under each parameter is shown in Table 2. "no strategy" means that no denoising strategy is applied to the audios. The best performance for each set is labeled as **red**. In this paper, the best performance for *set*0 is set to be the performance while no strategy is applied because clean audios are reference, although the performance under other configurations maybe better.

3.2.2. Local smoothing

Local smoothing replaces the value of a sample point with a smoothed value. This smoothed value can be the median value, the mean value, or other statistic values of a sliding window which is composed of $k-1$ points before and after the sample point. This paper chooses the median value, and we call local smoothing using median value *median smoothing* for simplicity. These transformations are not only useful, but also fast to operate and easy to implement. For median smoothing, the searching space of k ranges

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Table 3
Performance of median smoothing.

k	set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98
3	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
4	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
5	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
6	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
7	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
8	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
9	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
10	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48

from 3 to 10 with the step of 1. Table 3 shows the experimental results.

From Tables 2 to 3 we can observe that for each denoising strategy, each set of audios have a best configuration point. When the configuration approaches this point, the performance gets better; when the configuration departs from this point, the performance gets worse. It is observed that there is not a strategy that works the best for all classes of audios. For example, using median smoothing, for adversarial examples in *set*1, *set*2, and *set*3, the best configuration is $k = 4$, $k = 4$, and $k = 8$ respectively.

4. Methodology

Both quantization and median smoothing denoise the audio signal in the original domain, which does not take into account the frequency content. Therefore, we apply wavelet filter which denoises the audio signal in the transformed frequency domain. This section first leverages wavelet filter to mitigate adversarial perturbation for each class of audios, and then explores the performance of the combinations constructed by the single denoising strategies. Finally, we propose an intelligent noise reduction method.

4.1. Wavelet filter

Quantization and median smoothing denoise the signal in the original signal domain, wavelet filter denoises the signal in the transform domain. Wavelet denoising contains three steps: wavelet transform, coefficient denoising, and signal reconstruction. Wavelet transform transforms the input from the original signal domain to the wavelet domain. The signal is processed using a time-scale representation technique, which decomposes the signal into wavelet coefficients with different scales at different locations. Wavelet transform is implemented with the help of the correlation with translation and dilation of mother wavelet [44] such as Daubechies wavelets, symlets, coiflets, and so on. In coefficient denoising, the wavelet coefficients are further operated to remove the small coefficients which is assumed as noise. Common used denoising method is thresholding. There are two kinds of thresholding: hard thresholding and soft thresholding, which are expressed as follows:

$$Hard\ thresholding: \begin{cases} y = x & if|x| > \lambda \\ y = 0 & if|x| < \lambda \end{cases} \tag{8}$$

$$Soft\ thresholding: \{y = sign(x)(|x| - \lambda) \tag{9}$$

Hard thresholding may be too sensitive to small changes in the signal and thus is unstable, therefore, we choose soft thresholding in the denoising step. Finally, the denoised signal can be formed using reverse transform from the noise free coefficients.

Similar to quantization and median smoothing, the parameters of wavelet filter is also calibrated. However, the searching space of wavelet filter's parameter is much larger, since there are two parameters: the decomposition level–n and the wavelet name–*wname*. The denoising performance of wavelet filter will be shown in Section 5.

4.2. Combination of denoising strategies

Previous works only applies quantization and median smoothing independently. To explore new possibilities to enhance denoising performance, these strategies are combined in this section. It is assumed that, one denoising strategy can reduce the noises that present some specific regularities, while another strategy can reduce noises that present other regularities. If two strategies are combined, the noises that present the first kind of regularities and the noises that present the second kind of regularities may be mitigated simultaneously.

Each combination contains two denoising strategies. We traverse the 6 combinations constructed by the 3 denoising strategies: quantization and median smoothing, median smoothing and quantization, quantization and wavelet filter, wavelet filter and quantization, median smoothing and wavelet filter, wavelet filter and median smoothing. For each combination, the former one is first applied to the original audio, and then the second one is applied to the denoised audio. For example, using combination of quantization and median smoothing to denoise an audio, quantization is first applied, and then median smoothing is applied to the quantized audio to obtain the final audio.

4.3. Intelligent noise reduction method

In our observation, we find that different strategies have different capabilities in mitigating adversarial perturbations of audios indifferent class. The combination of two strategies can improve the performance against one set, but in real situation, the ASR system does not know which set the input audio belongs to. Therefore, INOR, an intelligent noise reduction method, is proposed.

INOR applies the best denoising strategy for a coming audio. This is achieved by classifying the input audio. INOR is mainly composed of two steps: model training and transcription prediction. The first step obtains a classification model, and the second step obtains the transcription with the help of the classification model. The overview of INOR is shown in Fig. 5.

4.3.1. Audio classification

Audio classification first needs to train a classification model. Model training contains two steps: feature extraction and model training.

In feature extraction, we extract essential features from the audios that can be leveraged to predict which classification sets the audios belong to. Each set of audios maps to a best denoising strategy, which can be either a single one or a combination. Although we cannot obtain the ground-truth transcription, we can predict the classification set of an audio by comparing the CERs using different denoising strategies.

The CER between the original transcription and the denoised transcription using one denoising strategy is used as one feature. Referring to Fig. 5, each audio in the training set has one label and three features. Here, "Wvlt" is the abbreviation of Wavelet. The label–$\mathbf{l}$ is the classification. There are four indexes in the classifications: 0, 1, 2, 3. 0 stands for *set*0, 1 stands for *set*1, 2 stands for *set*2, and 3 stands for *set*3. These denoising strategies are chosen because the experimental result shows that they are the best denoising strategy for the corresponding set of audios. The features are $\{\mathbf{h_1}, \mathbf{h_2}, \text{and } \mathbf{h_3}\}$. Here, $\mathbf{h}_i = CER(\mathbf{t}_i, \mathbf{t}_0)$. $\mathbf{t_0}$, $\mathbf{t_1}$, $\mathbf{t_2}$, and $\mathbf{t_3}$ are the transcriptions of $\mathbf{x_0}$, $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{x_3}$. $\mathbf{x_0}$ is the original audio; $\mathbf{x_1}$ is the audio denoised by *Wavelet&Quantization*; $\mathbf{x_2}$ is the audio denoised by *Wavelet&Quantization*; $\mathbf{x_3}$ is the audio denoised by

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

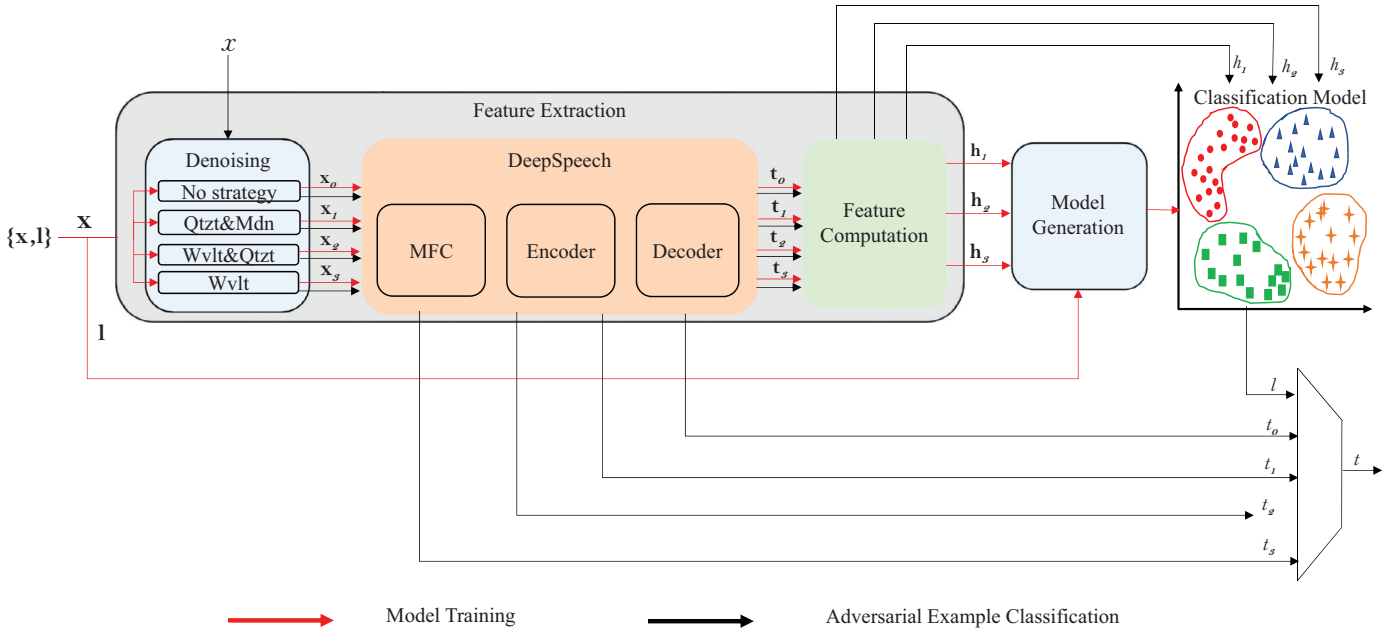


Fig. 5. Overview of INOR.

Quantization&Median. The features $\{\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_3}\}$, and their labels–**l** are leveraged to train a classification model. The labels are their classifications. The audios for generating the training dataset contain equal number of audios in each set. Since the number of features for each sample is comparatively small, the SVM model is selected.

4.3.2. Transcription prediction

After the classification model is obtained, when a new audio–x comes, the features-h_1, h_2, h_3 are firstly extracted. Then the audio is classified using the extracted features and the classification model. Finally, the transcription of the audio is determined according to the output of the classification label. If the label is 0, the transcription will be t_0; if the label is 1, the transcription will be t_1; if the label is 2, the transcription will be t_2; if the label is 3, the transcription will be be t_3. In this way, the audio is mapped to the best denoising strategy.

5. Experimental results

5.1. Wavelet filter and combinations

Wavelet filter and combinations are evaluated on each set of audios generated in Section 3.1. Table 4 presents the denoising performance using wavelet filter. In the experiment, n ranges from 1 to 5, with the step of 1, and *wname* ranges from *db*1 to *db*5, totally 25 configurations. Compared with quantization and median smoothing, wavelet shows great advantage in denoising the audios in *set*3, but for audios in *set*1 and *set*2 it is not the best.

The performance of the combinations are shown in Tables 5–7. Using combining strategies, the denoising performance for *set*1 and *set*2 get improved. The WER and CER for *set*1 is decreased to 0.13 and 0.05, and the WER and CER for *set*2 is decreased to 0.35 and 0.18. The sequence of the applied strategies affects the performance. Another obvious phenomenon is that, the performance is usually sensitive to one of the two denoising strategies in the combination. For example, using quantization-median smoothing, the WER and CER changes only when q changes. When q does not change and k ranges from 3 to 10, the WER and CER stays the same. The performance of other combinations also presents the same trend. This may be because that, under direct combination, one strategy is dominant, while the other one makes very small contribution.

Table 4
Performance of wavelet denoising.

$n-wname$	set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98
1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53
1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56
1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56
1-db4	0.31	0.15	0.19	0.08	0.42	0.23	0.82	0.56
1-db5	0.32	0.15	0.18	0.08	0.43	0.23	0.82	0.56
2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39
2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36
2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37
2-db4	0.32	0.15	0.26	0.11	0.43	0.21	0.65	0.38
2-db5	0.32	0.15	0.27	0.12	0.43	0.22	0.65	0.38
3-db1	0.39	0.20	0.35	0.18	0.44	0.23	0.60	0.35
3-db2	0.36	0.17	0.31	0.14	0.40	0.20	0.57	0.33
3-db3	0.33	0.16	0.28	0.13	0.44	0.22	0.58	0.32
3-db4	0.33	0.16	0.31	0.14	0.42	0.21	0.60	0.34
3-db5	0.34	0.16	0.32	0.15	0.42	0.22	0.61	0.35
4-db1	0.37	0.19	0.34	0.17	0.44	0.23	0.57	0.34
4-db2	0.35	0.16	0.31	0.15	0.42	0.22	0.57	0.33
4-db3	0.33	0.16	0.31	0.15	0.46	0.23	0.57	0.33
4-db4	0.34	0.16	0.33	0.15	0.43	0.23	0.58	0.34
4-db5	0.34	0.16	0.34	0.16	0.47	0.24	0.61	0.35
5-db1	0.37	0.19	0.34	0.17	0.45	0.24	0.58	0.34
5-db2	0.34	0.16	0.31	0.14	0.40	0.20	0.57	0.32
5-db3	0.34	0.16	0.30	0.14	0.45	0.22	0.59	0.32
5-db4	0.34	0.16	0.32	0.15	0.43	0.22	0.59	0.34
5-db5	0.34	0.16	0.30	0.14	0.45	0.24	0.62	0.34

5.2. INOR

The evaluation audios for INOR are the audios generated in Section 3.1. To train the classification model in INOR, 400 audios different from evaluation audios are used. The performance of INOR and the comparison with other denoising strategies is

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Table 5
Performance of quantization-median combination.

q, *k*	quantization–median								median–quantization							
	set0		set1		set2		set3		set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98
2,3	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,4	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,5	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,6	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,7	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,8	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,9	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
2,10	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74
3,3	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,4	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,5	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,6	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,7	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,8	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,9	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
3,10	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57	0.36	0.19	0.26	0.12	0.46	0.26	0.80	0.57
4,3	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,4	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,5	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,6	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,7	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,8	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,9	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
4,10	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47
5,3	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,4	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,5	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,6	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,7	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,8	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,9	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
5,10	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46
6,3	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,4	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,5	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,6	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,7	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,8	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,9	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
6,10	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.35	0.55	0.33	0.60	0.36	0.72	0.47
7,3	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,4	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,5	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,6	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,7	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,8	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,9	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
7,10	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54	0.68	0.46	0.71	0.47	0.71	0.48	0.77	0.54
8,3	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,4	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,5	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,6	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,7	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,8	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,9	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
8,10	0.80	0.57	0.79	0.57	0.79	0.58	0.83	0.61	0.83	0.63	0.83	0.63	0.84	0.64	0.87	0.66
9,3	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,4	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,5	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,6	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,7	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,8	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,9	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
9,10	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78	0.92	0.77	0.92	0.78	0.91	0.77	0.93	0.78
10,3	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,4	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,5	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,6	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,7	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,8	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,9	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87
10,10	0.96	0.85	0.95	0.85	0.96	0.84	0.96	0.85	0.96	0.87	0.97	0.87	0.96	0.87	0.96	0.87

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Table 6
Performance of median-wavelet combination.

$k, n-wname$	median–wavelet								wavelet–median							
	set0		set1		set2		set3		set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98
3,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
3,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
3,1-db5	0.32	0.15	0.18	0.08	0.43	0.23	0.82	0.56	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
3,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
3,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
3,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60
4,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
4,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
4,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
4,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
4,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
4,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51
5,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
5,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
5,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
5,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
5,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
5,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.41	0.22	0.31	0.16	0.46	0.26	0.72	0.46
6,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
6,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
6,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
6,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
6,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
6,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.41	0.21	0.34	0.17	0.49	0.27	0.73	0.45
7,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
7,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
7,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
7,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
7,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
7,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.47	0.26	0.41	0.22	0.53	0.30	0.73	0.46
8,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
8,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
8,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
8,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
8,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
8,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45
9,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
9,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
9,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
9,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
9,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
9,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.54	0.30	0.51	0.29	0.61	0.35	0.77	0.49
10,1-db1	0.33	0.17	0.21	0.09	0.45	0.24	0.77	0.53	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48
10,1-db2	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48
10,1-db3	0.32	0.15	0.19	0.08	0.42	0.23	0.83	0.56	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48
10,2-db1	0.37	0.19	0.30	0.14	0.43	0.23	0.65	0.39	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48
10,2-db2	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48
10,2-db3	0.32	0.15	0.25	0.11	0.41	0.21	0.63	0.37	0.58	0.33	0.55	0.30	0.64	0.36	0.76	0.48

shown in Table 8. This table includes the performance of single denoising strategies and combining denoising strategies. For each strategy, only the configuration that performs the best for a set of adversarial examples are presented. For example, using quantization, $q = 3$ achieves the best performance for *set*1, and $q = 6$ achieves the best performance for *set*2, while $q = 8$ achieves the best performance for *set*3. "average" means the average performance for all the audios included in all the sets.

On the whole, *quantization&median* with the configuration $q = 2, k = 3$ performs the best for *set*1; *wavelet&quantization* with the configuration $q = 4, n = 2, wname = db6$ performs the best for *set*2; *wavelet* with the configuration $n = 5, wname = db2$ and INOR performs the best for *set*3. In INOR, these three denoising strategies also take these configurations. INOR achieves the balance between the denoising performances for all the sets of groups. Comparing the average performance, INOR performs the best, with the WER and CER decreasing 33% and 55% from the performance while no strategy is applied. INOR is not only better than other strategies in total, but also makes a balance between the audios with different level of perturbations. For example, although quantization&median smoothing with $q = 2, k = 3$ achieves the best performance for audios in *set*1, the ER for set2 and set3 is too high, especially *set*3.

Besides the low error rate, INOR also successfully recover the semantics of all the adversarial examples in *set*1. Take the first example of *set*1 in Fig. 1 as an example, after denoised by INOR, the transcription is recovered to the clean transcription *"Without the dataset the article is useless"* from the adversarial transcription *"Without the dataset the article is not useless"*. For other adversarial examples in *set*1, the recovered transcription may not be so accurate, but all the deleted or added word "not" is recovered, which avoids great change in semantics.

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Table 7
Performance of quantization-wavelet combination.

$q, n - wname$	quantization–wavelet								wavelet–quantization							
	set0		set1		set2		set3		set0		set1		set2		set3	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
no strategy	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98	0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98
2,1-db1	0.34	0.17	0.21	0.09	0.44	0.24	0.78	0.53	0.33	0.16	0.22	0.09	0.43	0.23	0.80	0.54
2,1-db2	0.34	0.15	0.17	0.07	0.42	0.23	0.80	0.56	0.33	0.15	0.18	0.08	0.41	0.23	0.83	0.57
2,1-db3	0.32	0.15	0.18	0.08	0.41	0.23	0.82	0.55	0.31	0.14	0.18	0.07	0.41	0.22	0.84	0.57
2,1-db4	0.31	0.14	0.19	0.08	0.42	0.23	0.82	0.55	0.32	0.15	0.19	0.08	0.41	0.22	0.83	0.58
2,1-db5	0.31	0.14	0.18	0.08	0.44	0.23	0.81	0.55	0.32	0.14	0.19	0.08	0.43	0.23	0.82	0.57
2,1-db6	0.31	0.14	0.20	0.08	0.44	0.23	0.80	0.54	0.32	0.15	0.18	0.07	0.43	0.23	0.82	0.57
2,1-db7	0.31	0.14	0.20	0.09	0.44	0.24	0.82	0.54	0.32	0.14	0.20	0.08	0.42	0.23	0.84	0.57
2,1-db8	0.32	0.14	0.21	0.09	0.44	0.23	0.81	0.54	0.32	0.14	0.20	0.08	0.42	0.23	0.85	0.58
2,2-db1	0.38	0.19	0.31	0.14	0.42	0.22	0.64	0.38	0.37	0.19	0.31	0.14	0.43	0.23	0.65	0.38
2,2-db2	0.33	0.16	0.23	0.11	0.38	0.20	0.63	0.36	0.33	0.15	0.25	0.11	0.38	0.20	0.65	0.37
2,2-db3	0.33	0.15	0.26	0.11	0.41	0.21	0.64	0.37	0.33	0.15	0.25	0.11	0.38	0.19	0.68	0.39
2,2-db4	0.31	0.15	0.26	0.11	0.42	0.21	0.66	0.38	0.32	0.14	0.25	0.10	0.40	0.20	0.67	0.38
2,2-db5	0.32	0.15	0.26	0.12	0.42	0.22	0.64	0.37	0.32	0.15	0.25	0.11	0.38	0.20	0.67	0.38
2,2-db6	0.32	0.15	0.29	0.13	0.44	0.23	0.65	0.38	0.32	0.15	0.25	0.11	0.38	0.20	0.66	0.38
2,2-db7	0.32	0.16	0.29	0.13	0.45	0.23	0.68	0.40	0.31	0.15	0.25	0.11	0.39	0.20	0.66	0.38
2,2-db8	0.34	0.16	0.29	0.13	0.43	0.23	0.67	0.40	0.31	0.15	0.25	0.11	0.39	0.21	0.67	0.39
3,1-db1	0.33	0.17	0.20	0.09	0.43	0.23	0.77	0.52	0.34	0.17	0.22	0.09	0.42	0.22	0.78	0.52
3,1-db2	0.33	0.15	0.18	0.07	0.41	0.23	0.80	0.55	0.32	0.15	0.20	0.08	0.41	0.23	0.81	0.55
3,1-db3	0.32	0.15	0.19	0.08	0.41	0.22	0.81	0.55	0.32	0.15	0.18	0.07	0.40	0.22	0.82	0.55
3,1-db4	0.32	0.15	0.19	0.08	0.41	0.22	0.81	0.54	0.32	0.15	0.19	0.08	0.41	0.22	0.82	0.56
3,1-db5	0.32	0.15	0.18	0.08	0.42	0.23	0.82	0.54	0.31	0.14	0.21	0.08	0.40	0.21	0.82	0.56
3,1-db6	0.31	0.15	0.19	0.08	0.42	0.23	0.82	0.55	0.32	0.14	0.20	0.08	0.41	0.22	0.84	0.56
3,1-db7	0.32	0.15	0.20	0.08	0.43	0.23	0.81	0.54	0.32	0.14	0.20	0.08	0.41	0.22	0.84	0.57
3,1-db8	0.33	0.15	0.20	0.08	0.44	0.23	0.80	0.53	0.33	0.14	0.19	0.08	0.41	0.22	0.83	0.57
3,2-db1	0.39	0.20	0.30	0.14	0.42	0.23	0.64	0.39	0.39	0.19	0.32	0.14	0.42	0.22	0.64	0.37
3,2-db2	0.34	0.16	0.24	0.11	0.38	0.20	0.63	0.36	0.33	0.15	0.24	0.10	0.38	0.19	0.64	0.36
3,2-db3	0.33	0.16	0.26	0.11	0.41	0.21	0.64	0.37	0.34	0.15	0.25	0.10	0.38	0.19	0.64	0.37
3,2-db4	0.33	0.15	0.28	0.12	0.42	0.21	0.65	0.38	0.32	0.15	0.26	0.11	0.38	0.20	0.65	0.36
3,2-db5	0.33	0.16	0.27	0.12	0.42	0.22	0.65	0.37	0.32	0.15	0.24	0.10	0.37	0.19	0.65	0.37
3,2-db6	0.34	0.16	0.30	0.13	0.44	0.22	0.66	0.38	0.33	0.15	0.26	0.11	0.38	0.19	0.67	0.37
3,2-db7	0.34	0.17	0.29	0.13	0.44	0.23	0.66	0.40	0.31	0.15	0.25	0.10	0.37	0.19	0.65	0.37
3,2-db8	0.35	0.17	0.29	0.13	0.43	0.23	0.67	0.40	0.32	0.15	0.24	0.10	0.38	0.20	0.65	0.37
4,1-db1	0.34	0.18	0.21	0.09	0.43	0.22	0.76	0.50	0.35	0.18	0.22	0.10	0.42	0.22	0.76	0.48
4,1-db2	0.33	0.16	0.17	0.08	0.41	0.22	0.77	0.52	0.31	0.15	0.20	0.08	0.41	0.21	0.78	0.52
4,1-db3	0.33	0.16	0.19	0.08	0.42	0.21	0.77	0.50	0.32	0.15	0.20	0.08	0.40	0.21	0.78	0.52
4,1-db4	0.33	0.16	0.21	0.08	0.40	0.21	0.79	0.51	0.32	0.15	0.21	0.08	0.39	0.21	0.78	0.53
4,1-db5	0.32	0.15	0.20	0.08	0.42	0.22	0.78	0.51	0.31	0.15	0.20	0.08	0.41	0.21	0.81	0.54
4,1-db6	0.32	0.15	0.22	0.09	0.43	0.23	0.77	0.51	0.30	0.15	0.21	0.09	0.41	0.22	0.81	0.53
4,1-db7	0.32	0.15	0.21	0.09	0.42	0.22	0.78	0.51	0.31	0.14	0.20	0.08	0.40	0.21	0.81	0.53
4,1-db8	0.32	0.15	0.22	0.09	0.43	0.23	0.79	0.52	0.31	0.14	0.20	0.08	0.42	0.22	0.81	0.53
4,2-db1	0.38	0.19	0.31	0.15	0.43	0.23	0.65	0.39	0.39	0.20	0.33	0.16	0.42	0.22	0.61	0.36
4,2-db2	0.34	0.17	0.26	0.12	0.39	0.20	0.64	0.36	0.33	0.16	0.24	0.11	0.38	0.19	0.61	0.35
4,2-db3	0.33	0.16	0.27	0.12	0.40	0.20	0.64	0.37	0.35	0.16	0.23	0.10	0.37	0.18	0.63	0.36
4,2-db4	0.34	0.16	0.27	0.12	0.43	0.21	0.64	0.38	0.33	0.16	0.24	0.10	0.37	0.19	0.64	0.36
4,2-db5	0.35	0.16	0.27	0.12	0.41	0.21	0.63	0.37	0.31	0.15	0.22	0.10	0.36	0.18	0.63	0.36
4,2-db6	0.34	0.16	0.30	0.13	0.43	0.22	0.66	0.38	0.31	0.15	0.24	0.10	0.35	0.18	0.64	0.36
4,2-db7	0.34	0.16	0.30	0.13	0.46	0.24	0.64	0.39	0.31	0.15	0.24	0.10	0.34	0.18	0.64	0.36
4,2-db8	0.35	0.17	0.28	0.13	0.44	0.23	0.66	0.39	0.30	0.14	0.24	0.10	0.37	0.19	0.65	0.36
5,1-db1	0.37	0.19	0.25	0.12	0.44	0.23	0.72	0.46	0.38	0.20	0.26	0.12	0.45	0.23	0.71	0.44
5,1-db2	0.35	0.17	0.22	0.10	0.41	0.22	0.73	0.47	0.33	0.17	0.23	0.10	0.41	0.21	0.75	0.47
5,1-db3	0.36	0.17	0.24	0.11	0.42	0.22	0.75	0.46	0.32	0.16	0.24	0.11	0.40	0.22	0.74	0.47
5,1-db4	0.36	0.18	0.24	0.11	0.42	0.23	0.75	0.46	0.33	0.16	0.22	0.10	0.41	0.22	0.78	0.49
5,1-db5	0.36	0.17	0.23	0.10	0.41	0.22	0.72	0.46	0.32	0.16	0.22	0.10	0.41	0.21	0.76	0.49
5,1-db6	0.36	0.17	0.23	0.10	0.42	0.22	0.74	0.46	0.33	0.17	0.22	0.10	0.42	0.22	0.76	0.48
5,1-db7	0.36	0.17	0.23	0.10	0.42	0.22	0.74	0.46	0.34	0.17	0.22	0.10	0.43	0.22	0.78	0.49
5,1-db8	0.35	0.17	0.23	0.10	0.42	0.22	0.73	0.46	0.33	0.16	0.23	0.10	0.42	0.22	0.77	0.49
5,2-db1	0.41	0.21	0.34	0.18	0.44	0.24	0.62	0.38	0.40	0.21	0.37	0.19	0.44	0.24	0.64	0.37
5,2-db2	0.37	0.19	0.28	0.13	0.41	0.22	0.60	0.35	0.36	0.18	0.29	0.13	0.42	0.21	0.63	0.35
5,2-db3	0.36	0.17	0.29	0.14	0.41	0.21	0.62	0.36	0.35	0.17	0.27	0.12	0.39	0.20	0.63	0.36
5,2-db4	0.37	0.18	0.29	0.13	0.43	0.22	0.61	0.36	0.33	0.16	0.28	0.12	0.39	0.20	0.65	0.36
5,2-db5	0.36	0.17	0.29	0.13	0.43	0.22	0.64	0.37	0.33	0.17	0.27	0.12	0.39	0.20	0.64	0.36
5,2-db6	0.36	0.18	0.32	0.15	0.45	0.24	0.63	0.36	0.33	0.16	0.25	0.11	0.37	0.19	0.64	0.35
5,2-db7	0.37	0.19	0.32	0.15	0.46	0.25	0.64	0.38	0.34	0.16	0.28	0.13	0.39	0.20	0.63	0.36
5,2-db8	0.36	0.17	0.32	0.15	0.46	0.24	0.66	0.38	0.34	0.17	0.25	0.11	0.39	0.19	0.62	0.36

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Table 8
Performance of INOR.

strategy	parameter	set0		set1		set2		set3		average	
		WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
no strategy		0.30	0.14	0.18	0.11	0.54	0.43	1.19	0.98	0.55	0.42
quantization	q=3	0.32	0.15	0.14	0.06	0.52	0.37	1.09	0.91	0.52	0.37
	q=6	0.40	0.20	0.28	0.13	0.46	0.27	0.84	0.57	0.50	0.30
	q=8	0.61	0.39	0.58	0.36	0.63	0.40	0.74	0.51	0.64	0.42
median smoothing	k=4	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51	0.44	0.25
	k=8	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45	0.56	0.32
wavelet filter	n=1, wname='db2'	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.43	0.25
	n=2, wname='db2'	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.40	0.20
	n=5, wname='db2'	0.34	0.16	0.31	0.14	0.40	0.20	0.57	0.32	0.41	0.21
quantization&median smoothing	q=2, k=3	0.28	0.13	0.13	0.05	0.48	0.30	0.94	0.75	0.61	0.31
	q=4, k=5	0.35	0.17	0.24	0.11	0.41	0.22	0.73	0.48	0.43	0.25
	q=6, k=3	0.53	0.30	0.49	0.28	0.55	0.32	0.69	0.45	0.57	0.34
median smoothing&quantization	q=2, k=4	0.29	0.13	0.14	0.06	0.46	0.29	0.94	0.74	0.45	0.31
	q=2, k=3	0.35	0.18	0.25	0.12	0.42	0.23	0.71	0.47	0.43	0.25
	q=5, k=3	0.49	0.26	0.38	0.21	0.49	0.29	0.72	0.46	0.52	0.31
median smoothing&wavelet	k=9, n=2, wname='db2'	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.40	0.20
	k=10, n=1, wname='db2'	0.32	0.14	0.17	0.07	0.42	0.23	0.80	0.56	0.43	0.25
	k=10, n=2, wname='db2'	0.33	0.15	0.24	0.10	0.39	0.20	0.63	0.36	0.40	0.20
wavelet&median smoothing	k=3, n=1, wname='db1'	0.34	0.17	0.23	0.10	0.47	0.26	0.83	0.60	0.47	0.29
	k=4, n=1, wname='db1'	0.32	0.16	0.23	0.10	0.43	0.24	0.77	0.51	0.44	0.25
	k=8, n=1, wname='db1'	0.50	0.27	0.45	0.24	0.56	0.32	0.72	0.45	0.58	0.32
quantization&wavelet	q=2, n=1, wname='db2'	0.34	0.15	0.17	0.07	0.42	0.23	0.80	0.56	0.43	0.25
	q=2, n=2, wname='db2'	0.33	0.16	0.23	0.11	0.38	0.20	0.63	0.36	0.39	0.21
	q=5, n=2, wname='db2'	0.37	0.19	0.28	0.13	0.41	0.22	0.60	0.35	0.42	0.22
wavelet&quantization	q=2, n=1, wname='db3'	0.31	0.14	0.18	0.07	0.41	0.22	0.84	0.57	0.44	0.25
	q=4, n=2, wname='db2'	0.33	0.16	0.24	0.11	0.38	0.19	0.61	0.35	0.39	0.18
	q=4, n=2, wname='db6'	0.31	0.15	0.24	0.10	0.35	0.18	0.64	0.36	0.39	0.20
INOR		0.30	0.14	0.25	0.12	0.36	0.18	0.57	0.32	0.37	0.19

6. Conclusion

This paper analyzes the relationship between semantic difference and adversarial perturbations for adversarial audio examples. Based on the analysis, various denoising strategies, including traditional input transformation and wavelet filter are calibrated and their combinations are applied to mitigate the perturbations. It is observed that, for audios with different levels of transcription distance, the best denoising strategy differs. However, in practice, we do not know the level of transcription distance for a coming audio. Therefore, an intelligent noise reduction method–INOR is proposed to predict the level of transcription distance and further apply the most effective denoising strategy for the coming audio. Experimental results show that INOR is effective in mitigating the adversarial perturbations for all the groups of adversarial examples by decreasing the average CER and WER by 33% and 55% respectively.

Declaration of Competing Interest

The authors declare that they do not have any financial or non-financial conflict of interests.

Acknowledgment

This paper is supported in part by National Natural Science Foundation of China (NSFC) under grant Nos. (61532017, 61704174, 61432017, 61521092).

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NIPS, 2012, pp. 1097–1105.
[2] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 649–657.
[3] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 30–42.
[4] F. Yu, Z. Xu, C. Liu, X. Chen, Masker: adaptive mobile security enhancement against automatic speech recognition in eavesdropping, in: Proceedings of the Design Automation Conference (DAC), 2019, p. 163.
[5] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, D. Pérez-Cabo, No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation, IEEE Trans. Inf. Forensics Secur. 12 (11) (2017) 2640–2653.
[6] Z. Xu, F. Yu, C. Liu, X. Chen, HAMPER: high-performance adaptive mobile security enhancement against malicious speech and image recognition, Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC) (2019) 512–517.
[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing Properties of Neural Networks, arXiv:1312.6199(2013).
[8] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: Proceedings of the Symposium on Security and Privacy (SP), 2017, pp. 39–57.
[9] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, in: Proceedings of the International Conference on Learning Representations (ICLR), 2017.
[10] M. Sharif, S. Bhagavatula, L. Bauer, M.K. Reiter, Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1528–1540.
[11] I.J. Goodfellow J. Shlens C. Szegedy Explaining and Harnessing Adversarial Examples 2014 arXiv:1412.6572
[12] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
[13] J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, V. Fischer, Universal adversarial perturbations against semantic image segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2755–2764.
[14] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble Adversarial Training: Attacks and Defenses, arXiv:1705.07204 (2017).
[15] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: Proceedings of the Symposium on Security and Privacy (SP), 2016, pp. 582–597.
[16] J.H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations, in: Proceedings of the International Conference on Learning Representations, 2017.
[17] J. Lu, T. Issaranon, D.A. Forsyth, Safetynet: detecting and rejecting adversarial examples robustly, in: Proceedings of the International Conference on Computer Vision (ICCV), 2017, pp. 446–454.
[18] D. Meng, H. Chen, Magnet: a two-pronged defense against adversarial exam-

ples, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 135–147.
[19] Z. Xu, F. Yu, X. Chen, LanCe: A Comprehensive and Lightweight CNN Defense Methodology against Physical Adversarial Attacks on Embedded Multimedia Applications, arXiv Preprint arXiv:1910.08536 (2019).
[20] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M.E. Kounavis, D.H. Chau, Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning With Jpeg Compression, arXiv:1705.02900(2017).
[21] W. Xu, D. Evans, Y. Qi, Feature squeezing: detecting adversarial examples in deep neural networks, in: Proceedings of the Network and Distributed System Symposium (NDSS), 2018.
[22] Microsoft Cortana, https://www.microsoft.com/en-us/windows/cortana.
[23] Apple Sifi, https://www.apple.com/ios/siri.
[24] Google Now, https://www.androidcentral.com/google-now.
[25] Anazon Alexa, https://developer.amazon.com/alexa.
[26] P. Lamere, P. Kwok, W. Walker, E. Gouvea, R. Singh, B. Raj, P. Wolf, Design of the CMU sphinx-4 decoder, in: Proceedings of the European Conference on Speech Communication and Technology, 2003.
[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., The Kaldi speech recognition toolkit, in: Proceedings of the Workshop on Automatic Speech Recognition and Understanding, 2011.
[28] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al., Deep Speech: Scaling Up End-To-End Speech Recognition, arXiv:1412.5567(2014).
[29] M.M. Cisse, Y. Adi, N. Neverova, J. Keshet, Houdini: fooling deep structured visual and speech recognition models with adversarial examples, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2017, pp. 6977–6987.
[30] C. Kereliuk, B.L. Sturm, J. Larsen, Deep learning and music adversaries, IEEE Trans. Multimed. 17 (2015) 2059–2071.
[31] Y. Gong, C. Poellabauer, Crafting Adversarial Examples For Speech Paralinguistics Applications, arXiv:1711.03280(2017).
[32] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, W. Zhou, Hidden voice commands., in: Proceedings of the USENIX Security Symposium, 2016, pp. 513–530.
[33] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, W. Xu, Dolphinattack: inaudible voice commands, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 103–117.
[34] N. Carlini, D. Wagner, Audio adversarial examples: targeted attacks on speech–to-text, in: Proceedings of the Security and Privacy Workshops (SPW), 2018, pp. 1–7.
[35] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, C. Raffel, Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition, arXiv:1903.10346(2019).
[36] S. Sun, C.-F. Yeh, M. Ostendorf, M.-Y. Hwang, L. Xie, Training Augmentation With Adversarial Examples for Robust Speech Recognition, arXiv:1806.02782(2018).
[37] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, Y. Bengio, Invariant Representations for Noisy Speech Recognition, arXiv:1612.01928(2016).
[38] Z. Yang, B. Li, P.-Y. Chen, D. Song, Characterizing Audio Adversarial Examples Using Temporal Dependency, arXiv:1809.10875(2018).
[39] C. Ittichaichareon, S. Suksri, T. Yingthawornsuk, Speech recognition using MFCC, in: Proceedings of the International Conference on Computer Graphics, Simulation and Modeling (ICGSM), 2012, pp. 28–29.
[40] O. Viikki, K. Laurila, Cepstral domain segmental feature vector normalization for noise robust speech recognition, Speech Commun. 25 (1–3) (1998) 133–147.
[41] O. Temam, A defect-tolerant accelerator for emerging high-performance applications, in: Proceedings of the ISCA, 2012, pp. 356–367.
[42] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet Physics Doklady, 1966, pp. 707–710.
[43] D. Michelsanti, Z.-H. Tan, Conditional Generative Adversarial Networks for Speech Enhancement And Noise-Robust Speaker Verification, arXiv:1709.01703(2017).
[44] D. Novak, D.C. Frau, V. Eck, J.C. Pérez-Cortés, G. Andreu-García, Denoising electrocardiogram signal using adaptive wavelets, Energy (2000) 250.

Qingli Guo received the B.S. degree from Dalian Jiaotong University, Dalian, China, in 2014. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing. Her current research interests include IP protection, physical unclonable function, hardware security, and AI security.

Jing Ye received the B.S. degree in electronics engineering and computer science from Peking University, Beijing, China, in 2008 and the Ph.D. degree from the State Key Laboratory (SKL) of Computer Architecture (CA), Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2014. He is currently an Associate Professor with the SKL of CA, ICT, CAS. His current research interests include hardware security, AI security, physical unclonable function, hardware trojan, and very large scale integration testing and diagnosis.

Yiran Chen (M5SM6F8) received the B.S and M.S. degrees (Hons.) from Tsinghua University, Beijing, China, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2005. After five years in industry, he joined the University of Pittsburgh, Pittsburgh, PA, USA, in 2010, as an Assistant Professor, then promoted to an Associate Professor with tenure in 2014, and held Bicentennial Alumni Faculty Fellow. He is currently a tenured Associate Professor with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, where he serves as the Co-Director of the Duke Center for Evolutionary Intelligence. Dr. Chen was a recipient of five best paper awards and 15 best paper nominations from international conferences. He is a recipient of NSF CAREER award and ACM SIGDA outstanding new faculty award. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE DESIGN AND TEST OF COMPUTERS, the IEEE EMBEDDED SYSTEMS LETTERS, ACM Journal of Emerging Technologies in Computing Systems, and ACM Transactions on Cyber-Physical Systems, and served on the technical and organization committees of over 40 international conferences.

Yu Hu received the B.S., M.S., and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1997, 1999, and 2003, respectively, all in electrical engineering. She is currently a Professor with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her current research interests include autonomous navigation, autonomous driving, deep learning, and FPGA design.

Yazhu Lan received his Ph.D. degree from Chinese Academy of Sciences University, Beijing, in 2015. He is now an associate professor in Institute of Computing Technology, Chinese Academy of Sciences. From 2017, he has done postdoctoral research at Duke University, Durham, NC, USA. His current research interests include architecture for deep learning and the architecture based on FPGA. In addition, it has deep research on data center networks and core backbone architecture.

Guohe Zhang is now an associate professor in the school of Microelectronics at Xin Jiaotong University, Shaanxi, China. He received his B.S. degree in 2003 and Ph.D degree in 2008 in Electronics Science and Technology from Xin Jiaotong University, Shaanxi, China, respectively. In 2009, he joined the school of Electronic and Information Engineering as a lecturer. He was promoted to associated professor in 2013. From 2009 to 2011, he had a three year post-doctoral research in School of Nuclear Science and Technology from Xin Jiaotong University. From Feb to May of 2013, he had a short term visiting to the University of Liverpool, UK. His research interests fall in the area of Semiconductor Device Physics and Modeling, VLSI Design and Testing. Dr. Zhang co-authored more than thirty papers and applied more than twenty Chinese patents.

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110

Xiaowei Li received the B.Eng. and M.Eng. degrees in computer science from the Hefei University of Technology, Hefei, China, in 1985 and 1988, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 1991. He was an Associate Professor with the Department of Computer Science and Technology, Peking University, Beijing, from 1991 to 2000. In 2000, he joined ICT, CAS, as a Professor, where he is currently the Deputy Director of the State Key Laboratory of Computer Architecture. He has authored or co-authored over 300 papers in journals and international conferences, and holds 50 China patents and 50 software copyrights. His current research interests include very large scale integration testing, design for testability, dependable computing, hardware security, and wireless sensor networks. Dr. Li has been the Vice Chair of the IEEE Asia and Pacific Regional Test Technology Technical Council since 2004. He was the Chair of the Technical Committee on Fault Tolerant Computing, the China Computer Federation from 2008 to 2012, and the Steering Committee Chair of the IEEE Asian Test Symposium from 2011 to 2013. He was the Steering Committee Chair of the IEEE Workshop on RTL and High Level Testing from 2007 to 2010. He is currently a member of the Steering Committee of International Test Conference (ITC)-Asia. He is on the Technical Program Committee of several IEEE and ACM conferences, including ITC, VLSI Test Symposium, Design, Automation and Test in European, and Asia and South Pacific Design Automation Conference. He is an Associate Editor of Journal of Computer Science and Technology, Journal of Low Power Electronics, and Journal of Electronic Testing: Theory and Applications

Please cite this article as: Q. Guo, J. Ye and Y. Chen et al., INOR—An Intelligent noise reduction method to defend against adversarial audio examples, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.110