Neurocomputing 417 (2020) 357–370

Contents lists available at ScienceDirect

Neurocomputing

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

Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system ☆

Hyun Kwon [a], Hyunsoo Yoon [b], Ki-Woong Park [c,*]

[a] Department of Electrical Engineering, Korea Military Academy, Seoul 01819, South Korea
[b] School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[c] Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea

ARTICLE INFO

Article history:
Received 25 January 2020
Revised 14 May 2020
Accepted 31 July 2020
Available online 1 September 2020
Communicated by R. Capobianco Guido

Keywords:
Machine learning
Audio modification
Audio adversarial example
Defense technology
Deep neural network (DNN)

ABSTRACT

Deep neural networks (DNNs) display good performance in the domains of recognition and prediction, such as on tasks of image recognition, speech recognition, video recognition, and pattern analysis. However, adversarial examples, created by inserting a small amount of noise into the original samples, can be a serious threat because they can cause misclassification by the DNN. Adversarial examples have been studied primarily in the context of images, but their effect in the audio context is now drawing considerable interest as well. For example, by adding a small distortion to an original audio sample, imperceptible to humans, an audio adversarial example can be created that humans hear as error-free but that causes misunderstanding by a machine. Therefore, it is necessary to create a method of defense for resisting audio adversarial examples. In this paper, we propose an acoustic-decoy method for detecting audio adversarial examples. Its key feature is that it adds well-formalized distortions using audio modification that are sufficient to change the classification result of an adversarial example but do not affect the classification result of an original sample. Experimental results show that the proposed scheme can detect adversarial examples by reducing the similarity rate for an adversarial example to 6.21%, 1.27%, and 0.66% using low-pass filtering (with 12 dB roll-off), 8-bit reduction, and audio silence removal techniques, respectively. It can detect an audio adversarial example with a success rate of 97% by performing a comparison with the initial audio sample.

1. Introduction

Deep neural networks (DNNs) [1] provide excellent performance on classification problems and prediction problems. However, DNNs have a vulnerability because adversarial examples, created by inserting a little noise into the original sample, can cause misclassification by DNNs. For example, if an attacker adds optimized noise to a U-turn road sign, the modified road sign will still be correctly recognized as a U-turn sign in human perception but will be misrecognized as a left-turn sign by an autonomous DNN vehicle. Because such adversarial examples can cause misclassification by DNNs, a considerable amount of research on this issue is being conducted in the image domain.

Several studies on adversarial examples have presented audio domain approaches. Vaidya et al. [2] presented the "cocaine noodles" method, which can generate a mangled sound that cannot be understood by a person in order to mislead a speech recognition system. To improve the cocaine noodles method, Carlini et al. [3] proposed a hidden voice command to add human feedback, thereby improving the mangled sound that humans cannot understand. Zhang et al. [4] presented the dolphin attack, which causes a speech recognition system to be misled by generating a high-frequency sound outside the range of human hearing. Carlini and Wagner (CW) [5] proposed the CW attack method, which creates an audio adversarial example by adding a small amount of bit noise to the original sample. The CW method improves the connectionist temporal classification (CTC) loss function [6] by adding a small amount of bit noise to the original sample such that the result will not be mistaken by a human but will be mistaken by the speech recognition system. In response to the audio adversarial example attack methods described in these studies, additional studies on defense methods are needed as well.

☆ A preliminary version of this paper was presented at ACM CCS 2019.

* Corresponding author.
E-mail addresses: hkwon.cs@gmail.com (H. Kwon), hyoon@kaist.ac.kr (H. Yoon), woongbak@sejong.ac.kr (K.-W. Park).

https://doi.org/10.1016/j.neucom.2020.07.101
0925-2312/

In this paper, we propose an acoustic-decoy detection method that can decrease the effectiveness of a CW attack, which is a state-of-the-art attack on the DeepSpeech model [7]. Our method relies on the difference between the classification result of the original image and that of the adversarial example after applying audio modification. The key feature of the proposed method is that it adds well-formalized distortions using audio modification sufficient to induce a change in the classification result of the adversarial example (because of its sensitivity) but not so much that the classification result of the original sample will be affected. The method uses this feature to detect audio adversarial examples. This study is an extension of our previous work [8], presented at ACM CCS 2019, in which we focused on concepts and ideas for detecting an adversarial example. In the current study, we focus on the detection of audio adversarial examples by using audio modification. The contributions of this paper for defending against adversarial examples are as follows:

- We describe the principle and the procedure of the proposed method, and we systematically show its foundation. We experimentally demonstrate using a Mozilla dataset that the proposed method can be used for detecting adversarial examples.
- We analyze the spectra, waveforms, similarity rates, and detection rates that are produced by applying the proposed method. The proposed method is quantitatively compared with existing audio defense methods. We also present the possibility of combining the method with others to create various ensemble methods of audio modification.
- We show the performance of the proposed scheme for defending the state-of-the-art DeepSpeech model against an attack that uses the state-of-the-art CW method. In addition, we analyze the decibel difference between the original sample and adversarial examples after applying audio modifications using the low-pass filtering, 8-bit reduction, and audio silence removal techniques.

The rest of this paper is structured as follows: In Section 2, we describe related work and provide background information on the target speech recognition system and attacks using adversarial examples. The conceptual basis for the proposed scheme is given in Section 3. Section 4 introduces the proposed detection scheme. In Section 5, results of experiments using the proposed method are presented. The proposed method is discussed in Section 6. Finally, Section 7 concludes the paper.

2. Related work and background

Szegedy et al. [9] first proposed an adversarial example that can cause misclassification by a DNN classifier yet has minimal distortion from the original sample.

The structure of this section is as follows. Section 2.1 explains the target speech-to-text recognition system. In Sections 2.1–2.6, five aspects of adversarial examples are described: classification by target model information, classification by type of recognition intended, distortion measure, audio adversarial example attacks, and audio adversarial example defenses.

2.1. Targeted speech-to-text model

Hidden Markov models (HMMs) [10] predict label sequences of speech data after applying pre-segmentation and post-processing. However, this method exponentially increases the number of cases by 26^N per character, and so it is not feasible to calculate all of the possible phrases. In contrast with pre-segmentation and post-processing with HMMs, the connectionist temporal classification (CTC) method, presented by Graves et al. [11], uses recursive neural networks (RNNs) [12] to directly train an unsegmented sequence label. This method maximizes the number of correct labels per input sequence in a probability distribution among all label sequences. To enhance the scalability of the CTC, Hannun et al. [7] presented the DeepSpeech model, which is an optimized-RNN training system using multiple GPUs. The DeepSpeech model can provide synthesis methods to efficiently create a variety of data. In this study, we used the DeepSpeech model as the target speech-to-text model in our testing.

2.2. Information known about target model

In terms of the information known about the target model, adversarial examples can be divided into two types: black box attacks and white box attacks. In a black box attack, the attacker does not have the target model information but can query the target model. In a white box attack, on the other hand, the attacker has all of the information about the target model, such as the probabilities of the output class result. The scheme proposed in this paper assumes that the attacker does not have information about the detector but rather is executing a limited-knowledge attack, in which the attacker knows about the target classifier but does not know that a detector employing audio modification is being used.

2.3. Type of recognition intended

We can also divide adversarial examples according to the class as which they are intended to be recognized by the target model [13–15]; these two categories are untargeted adversarial examples and targeted adversarial examples. An untargeted adversarial example can cause the target model to recognize the adversarial example as any class other than the original class. A targeted adversarial example, on the other hand, is designed to cause the target model to recognize the adversarial example as a particular target class selected by the attacker. In this paper, the proposed method assumes a targeted adversarial example that can choose the target class.

2.4. Distortion measure

In the audio domain, the proposed method applies the L_∞ measure of distortion [13], denoted as δ. If the distortion δ (noise level) is expressed in decibels, then $dB(\delta) = \max_i\{20 \cdot \log_{10}(|x_i^* - x_i|)\}$ [5]. The smaller the value of $dB(\delta)$, the more similar the adversarial example is to the original sample.

2.5. Audio domain methods of adversarial example attack

Vaidya et al. [2] presented the "cocaine noodles" method, which can generate a mangled command sound that cannot be understood by a person in order to mislead a speech recognition system. After extracting each feature from the mel-frequency cepstral coefficients (MFCC) parameter, the method inverts the MFCC. This method can cause the malfunction of a machine without considering the distortion.

To enhance the cocaine noodles method, Carlini et al. [3] proposed a hidden voice command method to add human feedback to improve the strange sound that humans cannot understand. The hidden voice produced by the method is not comprehensible to human perception and can induce a malfunction. Like the cocaine noodles method, this method inverts the MFCC after extracting each feature from the MFCC parameter. The method then tests the mangled commands using human feedback. This

method is a version of the cocaine noodles method, extended to black box attacks by incorporating human recognition testing.

In contrast to methods that create hidden voice commands, Zhang et al. [4] presented the dolphin attack, which causes the speech recognition system to be misled by producing a high-frequency sound outside the range of human hearing. In addition, the Houdini method was suggested by Cisse et al. [16] for optimization using the CTC loss function. A targeted audio attack was proposed by Alzantot et al. [17] as a black box attack that operates by inserting background noise, which achieved an attack success rate of 87%. However, it is limited in that it only has ten classes in the datasets.

Recently, Carlini and Wagner [5] generated an audio adversarial example by inserting a slight noise into the original sample. The generated adversarial example was misclassified as the target phrase by the model. The Carlini method uses a modified CTC loss function to create an adversarial example:

$$\text{minimize } dB_x(x^*, x) + \sum_i c_i \times g_i(x^*, \pi^i), \tag{1}$$

where $\sum_i c_i \times g_i(x^*, \pi^i)$ is a loss function of the sequence [5]. $dB_x(x, x^*)$ is a distortion loss function between original sample x and adversarial example x^*. DeepSpeech misclassifies x^* as target phrase t because of this loss function of the sequence. By adjusting the value of c appropriately, the Carlini method creates an adversarial example that will be misinterpreted as the target phrase by the model, while minimizing the distortion. The experiments conducted in this study used the Carlini method, a state-of-the-art targeted audio attack, to create the adversarial examples.

2.6. Audio domain methods of adversarial example defense

Some studies of defense-related methods in the audio domain do not specifically address audio adversarial examples. The spoofing detection method [18] uses a Gaussian mixture model (GMM) and a deep neural network to detect spoofing attacks. Unlike conventional methods, it includes additional detection through scoring by humans in a log-likelihood method. Another method, the temporarily aware context modeling method [19], performs speech activity detection that guarantees temporary continuity to an expected signal using a generative adversarial net. This method determines whether an observed audio segment has salient information by predicting the audio sequence in the next frame.

Methods of defense in the audio domain that do address attack by adversarial examples include the white noise method, down-sampling, and the temporal dependency method. The first of these, proposed by Subramanian et al. [20], detects adversarial examples using white noise. This method adds white noise to the input data as a standard digital distortion. The second method, proposed by Tamura et al. [21], is a denoise down-sampling method with a sandbox approach. This method determines whether an input is an adversarial example or an original sample by randomly down-sampling the input data and removing low-frequency sounds. The temporal dependency method, proposed by Yang et al. [22], uses the property of temporal information loss in the original sequence due to the adversarial noise in the adversarial example. This method determines whether the input is an adversarial example or an original sample by comparing the concordance rate between the classification result of the k portion of the speech and the k portion of the entire classification result.

These last three defense methods were used in the performance analysis conducted in the present study for comparison with the proposed method (Section 5.2.5).

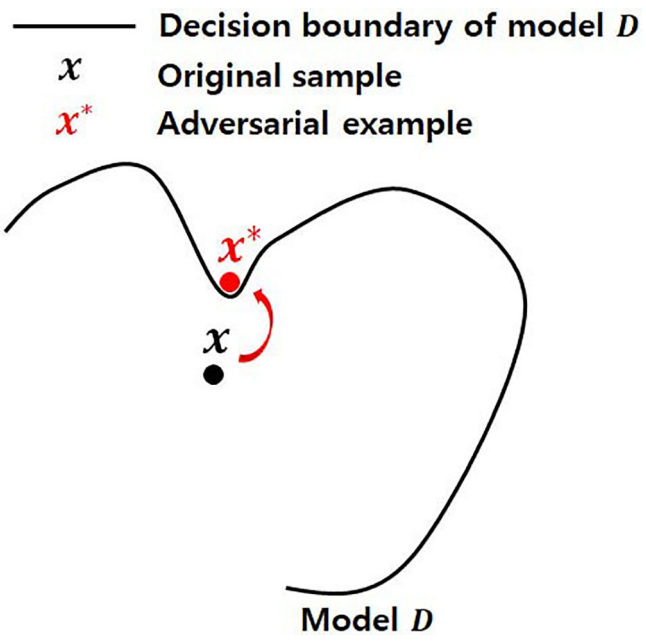


Fig. 1. Example of an adversarial example and its corresponding original sample in relation to the decision boundary of a target model.

3. Conceptual basis for proposed scheme

Fig. 1 shows an example of an adversarial example and its corresponding original sample in relation to the decision boundary of a model D. Model D correctly classifies the sample that is within the boundary. However, an adversarial example can be created just outside the boundary of model D. It is considered an adversarial example because it will be incorrectly classified by model D and yet is minimally distorted from the original sample.

In the figure, because the adversarial example is near the decision boundary, it is sensitive to class changes due to external distortion. On the other hand, as the original sample lies inside the decision boundary, even if the distance is changed by an external distortion, the original class will remain unaffected. Therefore, the adversarial example can be detected using its relatively greater sensitivity to external distortion.

4. Proposed scheme

4.1. Assumption

The proposed method assumes that the attacker does not have information about the audio modification used by the detector. Rather, it is a limited-knowledge attack: The attacker knows about the target classifier but does not know that a detector employing audio modification is being used. In other words, the model architecture, parameters, and probabilities of output classifications for the target classifier are known. Under this assumption, the attacker creates an optimized adversarial example with minimal distortion through multiple iterations, which causes misclassification by the target classifier. Therefore, the adversarial example has a high attack success rate against the target classifier, with minimal distortion from the original sample.

The target classifier is a neural network such as those used in artificial intelligence synthesizers [23], self-driving cars [24], speech classifiers [7], and many other applications [25,26].

4.2. Proposed method

Fig. 2 illustrates the concept of the proposed scheme, which comprises two procedures. First, the initial sample is verified against the recognition system and is given an initial classification result. Next, a modified sample is created by audio modification. Then, the classification result for this modified audio signal is compared with that for the initial audio sample. The difference between the result for the initial classification and that for the last classification plays a major role in the method because a given audio sample will be classified as either an original sample or an adversarial example based on how large the difference is. If there is a very large difference between the results, the sample will be classified as an adversarial example; if there is not so much difference, it will be classified as an original sample.

Expressed in mathematical terms, the first step is to calculate the classification results $f(t_{\text{initial}})$ for the input t_{initial} if the proposed method receives the original sample or the adversarial example as the input value. Then, input value t_{initial} is passed through modification module $g(t_{\text{initial}})$ to generate modified sample t_{modified}. The classification result $f(t_{\text{modified}})$ is calculated by the classifier using the modified sample t_{modified}.

In the second step, the coincidence rate is checked for the calculated classification results $f(t_{\text{modified}})$ and $f(t_{\text{initial}})$ to determine whether it is an adversarial example (if the rate is less than a predetermined threshold T) or an original sample (if the rate is greater).

Expressed in terms of the principle, the proposed method uses the features of the audio adversarial example. In the process of generating an adversarial example, distortion is inserted into the original sample to the point at which the machine begins to misinterpret the signal. Therefore, when the distortion is added by audio modification, the difference in classification for an adversarial example will be larger than that for an original sample.

As one audio modification technique, various types of filters can be applied, such as low-pass filters [27], high-pass filters [28], or notch filters [29]. An analysis of various audio modification techniques for use in the proposed method is given in Sections 5 and 6. As low-pass filtering, 8-bit reduction, and the removal of audio silence displayed good detection performance because of their minimal noise, which is due to the characteristics of the adversarial example, we use these three techniques for the audio modification. The details of the procedure for detecting an adversarial example using each audio modification technique are given in Algorithm 1.

Algorithm 1 Adversarial example detection

Input: test audio t_{initial}, classifier function $f(\cdot)$, modification function $g(\cdot)$, coincidence threshold T
Adversarial example detection:
  $result_{\text{before}} \leftarrow f(t_{\text{initial}})$
  $t_{\text{modified}} \leftarrow g(t_{\text{initial}})$
  $result_{\text{after}} \leftarrow f(t_{\text{modified}})$
  $w_{\text{d}} \leftarrow \text{coincidence}(result_{\text{after}}, result_{\text{before}})$
  if $w_{\text{d}} \leqslant T$ **then**
    $flag \leftarrow 1$
  else
    $flag \leftarrow 0$
  end if
  return flag

5. Experiment and evaluation

Through experiments, we show that the proposed method can effectively detect adversarial examples by using audio modification. We used the Tensorflow library [30], widely used for machine learning, and an Intel(R) i5-7100 3.90-GHz server. This section consists of two parts, the experimental setup and the experimental results.

5.1. Experimental setup

In the experiment, pre-trained model D essentially had the structure of DeepSpeech [7]. The initial seed for D was 1234, and the initial weight for D was 0.053424. The training data were from the Fisher, Switchboard, and Wall Street Journal corpora provided by the Linguistic Data Consortium [7]. Model D was pre-trained using 9600 training data [7] that consisted of 5000 h by 9600 speakers.

For test data, the Mozilla Common Voice dataset [31] was used, consisting of 100 arbitrary samples, which are described in Table 3 (Appendix). In the results on the original test data, model D showed an error rate of 16.49% on the test data. Given this error rate, DeepSpeech cannot interpret 100% of the test samples of the original sentence, but it can interpret whether a test sample is similar to the original test sentence. Fig. 3 shows the test result for the original sample "the boy went to his room and packed his belongings" as interpreted by model D. As can be seen in the figure, although there are some differences due to the error rate, model D interprets the original sample as being similar to the original sentence.

A demonstration of the performance of the proposed method must take into account the fact that the proposed method assumes that the attacker does not have information about the detector but rather is a limited-knowledge attack, in which the attacker knows about the target classifier but does not know that a detector employing audio modification is being used. To generate audio adversarial examples using the state-of-the-art Carlini–Wagner attack, Adam [32] was used as an optimizer, with a learning rate of 10. We created 100 randomly targeted adversarial examples against the target classifier.

5.2. Experimental results

The accuracy is the rate of coincidence between letters in the phrase output by D and letters in the original phrase; it is given by $(O-D-S-I)/O$, where O is the number of letters in the original sentence, D is the number of deletions, S is the number of substitutions, and I is the number of insertions. The similarity rate is the proportion of matches between the initial classification result (recognized before the audio modification) and the classification result recognized after the audio modification; it is given by $(B-U-N-E)/B$, where B is the number of letters in the initial classification result, U is the number of substitutions, N is the number of insertions, and E is the number of deletions. If there is no difference between the results of classification before and after the audio modification, the similarity rate will be higher; if the difference between the results of classification before and after the audio modification is large, the similarity rate will be lower. The definition of distortion δ is $dB(\delta) = \max_i\{20 \cdot \log_{10}(|x_i^* - x_i|)\}$.

5.2.1. Analysis of low-pass filtering

The low-pass filtering technique filters out sounds having a specific high frequency. The parameters of a low-pass filter are the cut-off frequency and roll-off (slope). The cut-off frequency is the boundary point between the frequency band through which a signal may pass and the frequency band through which signals cannot pass. The roll-off is the decreasing slope outside the cut-off frequency, which means the decibel decreases every octave. The higher the roll-off value, the steeper the reduction slope. In

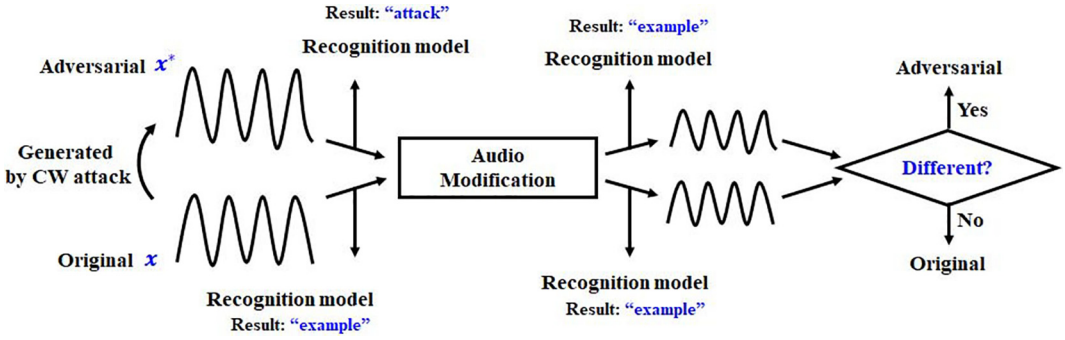


Fig. 2. Proposed architecture.

Original sentence:

"the boy went to his room and packed his belongings"

Model D transcription of the original sample:

"the boy went to his rom and paced his belonging"

Fig. 3. Example of transcription of an original sample by model *D*.

the low-pass filter used in the experiment, the cut-off frequency was set to 1000 Hz, and the roll-off was set to 6, 12, 24, 36, or 48 dB per octave.

Fig. 4 shows the waveforms before and after low-pass filtering for an original sample and a corresponding audio adversarial example. Fig. 4(b) shows that a small amount of bit noise has been added throughout the waveform shown in Fig. 4(a). In particular, referring to the spectra in Fig. 5, it can be seen that Fig. 4(b) shows overall noise, in contrast with Fig. 4(a). Therefore, the audio waveforms of Figs. 4 (a) and 4(b) are nearly identical. However, Fig. 6 shows that the recognition system correctly recognizes the audio of Fig. 4(a) as "an y going to tell me" but misrecognizes the audio of Fig. 4(b) as "example," as chosen by the attacker. On the other hand, the modified samples after the audio modification technique has been applied (Fig. 4(c) and (d)) are similar to the original sentence. Thus, for the original sample, the waveforms (Fig. 4(a) and (c)) are different but yield the same interpretation (Fig. 6). For the adversarial example, however, it can be seen (Fig. 6) that the interpretation of the waveform in Fig. 4(d) is similar to the original sentence, owing to the effects of the audio modification, which removes the adversarial noise.

We tested 100 samples and examined the concordance rate for the recognized sentence in the adversarial example and the original sample. Fig. 7 shows the similarity rates for the adversarial example and for the original sample over the decibel range of an octave in the low-pass filtering method. In Fig. 7, the similarity rate for the original sample is maintained for roll-off values up to 12, whereas the similarity rate for the adversarial example decreases. However, if the roll-off increases to greater than 24, the similarity rate for the original sample is reduced because of the severity of the audio modification. Therefore, at exactly 12 dB, it can be considered to have hit a "sweet spot.".

The audio files can be heard directly at the links given in [33,34] (original sample before and after audio modification) and [35,36] (audio adversarial example before and after audio modification).

5.2.2. Analysis of 8-bit reduction

The 8-bit reduction technique reduces a 16-bit audio sample to an 8-bit audio sample. Figs. 8 and 9 show the waveforms and spectra, respectively, of an original sample and a corresponding adversarial example before and after 8-bit reduction. First, we can see from Figs. 8(b) and 9(b) (before the application of the 8-bit reduction) that a little noise has been added overall compared with Figs. 8(a) and 9(a). However, as shown in Fig. 10, the original sample (before audio modification) of Fig. 8(a) is recognized as being the same as the original sentence, "this is for you," whereas the adversarial example (before audio modification) of Fig. 8(b) is misunderstood as "example," as chosen by the attacker. Next, we can see that after application of the 8-bit reduction (Fig. 8(c) and (d)), the number of samples (y-axis) is substantially reduced. This is because the number of bits in the audio file drops from 16 to 8 as a result of the 8-bit reduction. Fig. 9(c) and (d) show similar patterns throughout the spectra, but with additional noise overall. However, as shown in Fig. 10, the original sample (Fig. 8(c)) is recognized as the original sentence, "this is for youd," and the adversarial example (Fig. 8(d)) is also recognized as the original sentence, "tiso for youd," instead of "example" as chosen by the attacker. As can be seen, because 8-bit reduction reduces the effect of adversarial noise, the adversarial example is now correctly recognized as the original sentence. Compared with other techniques, 8-bit reduction produces greater distortion of the original sample, with the result that the similarity rate for the original sample is relatively low (Table 1).

The audio files can be heard directly at the links given in [37,38] (original sample before and after audio modification) and [39,40] (audio adversarial example before and after audio modification).

5.2.3. Analysis of audio silence removal

The audio silence removal technique removes unnecessary silence from sound. After it checks the beginning and end of the sample, unnecessary parts are removed by this technique. The

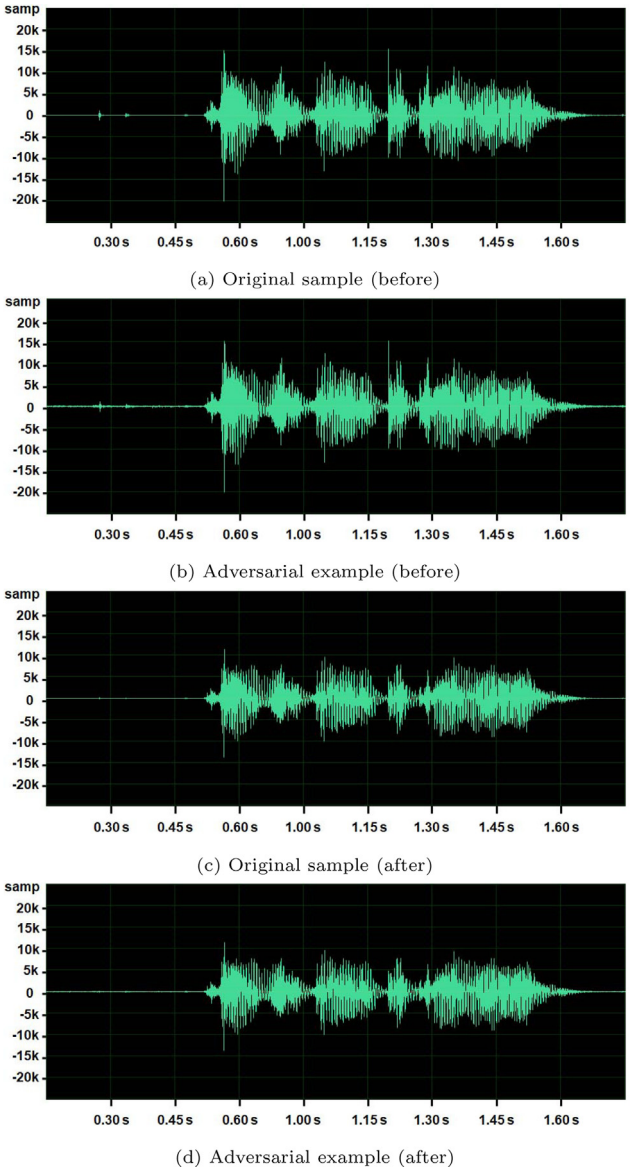


Fig. 4. Waveforms for an original sample and an audio adversarial example, before and after low-pass filtering. "Before" is before low-pass filtering; "after" is after low-pass filtering.

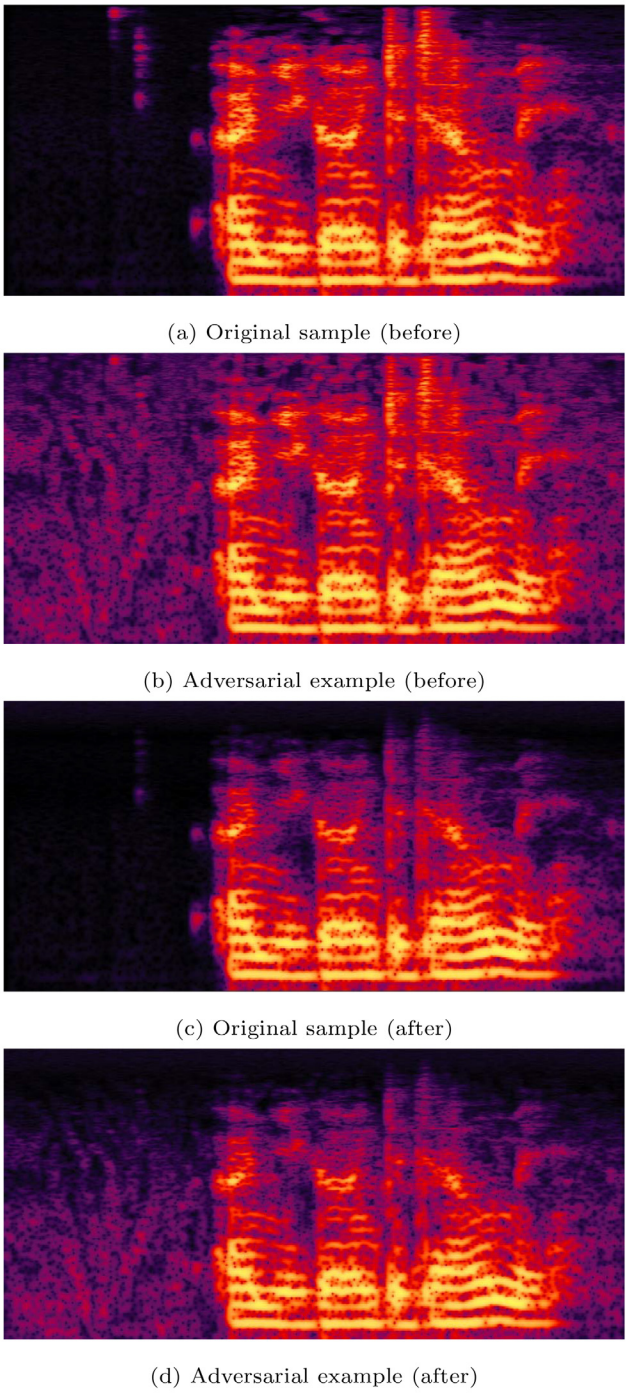


Fig. 5. Spectra for waveforms shown in Fig. 4. "Before" is before low-pass filtering; "after" is after low-pass filtering.

algorithms used for silence removal are the zero-crossing rate (ZCR) [41] and the short-time energy (STE) [42]. The ZCR algorithm is a voice activity detection method that uses the change point (from positive to negative or from negative to positive) of the sign function. The STE algorithm can effectively classify voice and non-voice segments by using the feature that the energy of voice sound is greater than that of non-voice sound. As the parameter values for the two algorithms, 16,000 Hz was used as the sampling rate, the threshold was −20 dB, and the window type was Hamming.

Figs. 11 and 12 show the waveforms and spectra, respectively, of an original sample and a corresponding adversarial example before and after audio silence removal. First, we can see from Figs. 11(b) and 12(b) (before the audio silence removal is applied) that a little noise has been added overall compared with Figs. 11(a) and 12(a). However, as shown in Fig. 13, the original sample (before audio modification) of Fig. 11(a) is recognized as being the same as the original sentence, "isn't the party also to announce his engagement to joanna," whereas the adversarial example (before audio modification) of Fig. 11(b) is misunderstood as "example," as chosen by the attacker. Next, we can see that after application of the audio silence removal (Figs. 11(c) and 11(d)), the silent part of each waveform has been removed. The runtimes have also decreased, from ∼ 4 s to ∼ 3 s. Figs. 12(c) and 12(d) also show similar patterns throughout the spectra, but the unnecessary parts have been removed. In terms of recognition, as shown in Fig. 13, the original sample (Fig. 11(c)) is recognized as the original sentence, "isn't the party also to announce his engagement to joanna," and the adversarial example (Fig. 8(d)) is also recognized as the original sentence, "isot party also to announce his engagt-ment to joanna," instead of "example" as chosen by the attacker.

Original sentence: "aren't you going to tell me"

Sentence recognized from Figure 4 (a) and (c): "an y going to tell me"

Sentence recognized from Figure 4 (b): "example"

Sentence recognized from Figure 4 (d): "nd e going to tell me"

Fig. 6. Sentences recognized by DeepSpeech from waveforms shown in Fig. 4.

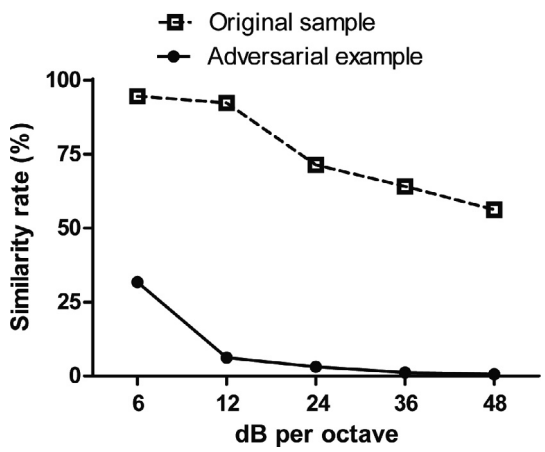


Fig. 7. Similarity rates for adversarial example and original sample through low-pass filtering.

It can be seen that the adversarial noise has been partially removed from the waveform throughout the audio and that the adversarial example after audio silence removal is recognized as the original sentence, "isot party also to announce his engagtment to joanna.".

The audio files can be heard directly at the links given in [43,44] (original sample before and after audio modification) and [45,46] (audio adversarial example before and after audio modification).

5.2.4. Comparisons of similarity rates, decibels, and detection rates

Table 1 shows the similarity rates for the original sample and for the adversarial example using the low-pass filtering, 8-bit reduction, and audio silence removal techniques. The low-pass filtering technique filters out sounds having a specific high frequency. In the case of the original sample under low-pass filtering, the similarity rate for the original sample was maintained at 93.94% because there is less voice corresponding to the high frequency in the original sample. However, in the case of the adversarial example under low-pass filtering, adversarial noise is reflected in the entire frequency band for generating an adversarial example, but a specific high frequency has been removed, and the difference between the classification results before and after the audio modification is large. Therefore, the similarity rate for the adversarial example was reduced to 6.21%.

The 8-bit reduction technique reduces a 16-bit audio sample to an 8-bit audio sample. If the.wav files input to DeepSpeech are 8-bit files, DeepSpeech's recognition rate drops. This is because the DeepSpeech model is optimized to correctly recognize.wav files that are 16-bit samples taken at 16 kHz. Therefore, for an original sample under 8-bit reduction, the difference between the classification results before and after the audio modification is relatively large, and so the similarity rate for the original sample was low, 57.75%. In the case of an adversarial example, some of

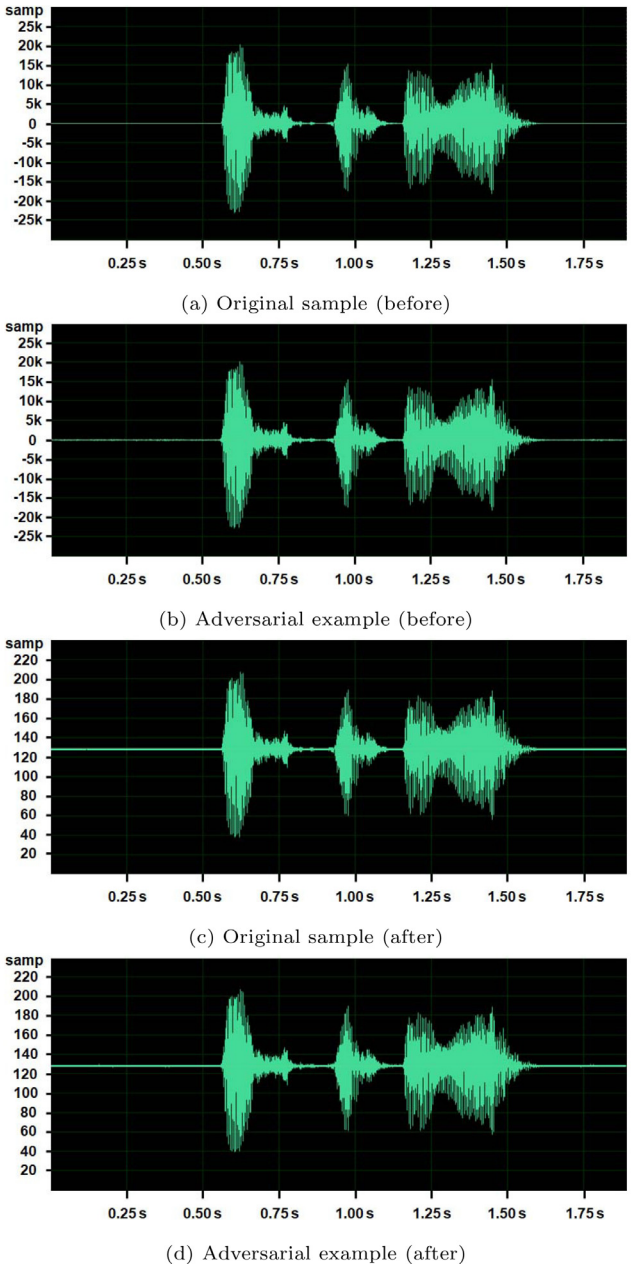
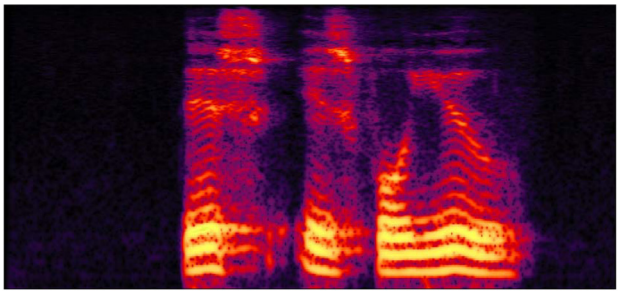
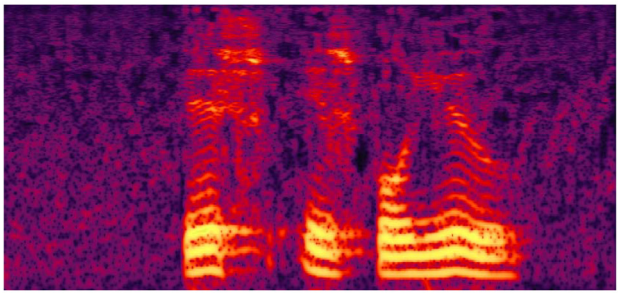


Fig. 8. Waveforms for an original sample and an audio adversarial example, before and after 8-bit reduction. "Before" is before 8-bit reduction; "after" is after 8-bit reduction.

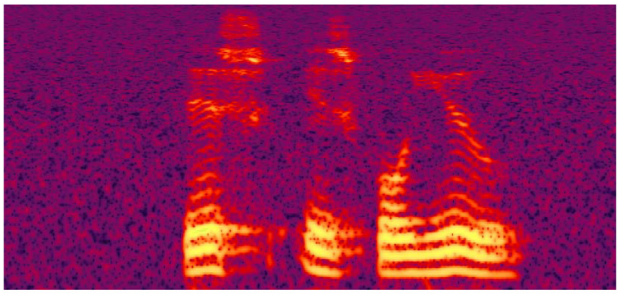
the adversarial noise in the adversarial example is lost by the 8-bit reduction, and the input value has changed to 8 bits; thus, the difference between the classification results before and after



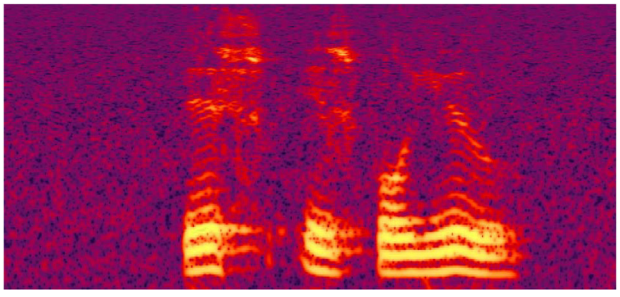
(a) Original sample (before)



(b) Adversarial example (before)



(c) Original sample (after)



(d) Adversarial example (after)

Fig. 9. Spectra for waveforms shown in Fig. 8. "Before" is before 8-bit reduction; "after" is after 8-bit reduction.

the audio modification is large. Therefore, the similarity rate for the adversarial example was reduced to 1.27%.

The audio silence removal technique removes unnecessary silence from sound. In the case of an original sample, after the silence segmentation, the speech recognition is less accurate because the endpoints are cut off or because voices frequently overlap. Therefore, there is a slight difference in the classification results before and after the audio modification, and the similarity rate for the original sample was 71.44%. In the case of an adversarial example, on the other hand, because the optimized adversarial noise of the adversarial example is partially deleted by the silence removal, the difference between the classification results before and after the audio modification is large. Therefore, the similarity rate for the adversarial example was reduced to 0.66%.

In terms of the similarity rate for the original sample, it can be seen from the table that the low-pass filtering technique produces a higher similarity rate than the other techniques and that the 8-bit reduction technique produces a much lower similarity rate, 57.75%. In terms of the similarity rate for the adversarial example, audio silence removal produces a lower similarity rate than the other techniques. It can be seen that the adversarial example is recognized correctly by the removal of the adversarial noise as reflected in the silences. We can see that the performance of the audio silence removal technique is better than 8-bit reduction in terms of both the similarity rate for the original sample and the similarity rate for the adversarial example. However, audio silence removal has a lower similarity rate on the adversarial example than low-pass filtering but also a lower similarity rate on the original sample than low-pass filtering.

Fig. 14 shows the decibels and the difference in the decibels for the original sample and the adversarial example using the low-pass filtering (6, 12, 24, and 48 dB), 8-bit reduction, and audio silence removal techniques applied to 100 test data. As seen in the figure, the adversarial example and the original sample show little difference in terms of decibels. On the other hand, in terms of the difference in the decibels for each technique, the difference under the low-pass filtering technique decreases because this technique removes the least noise from the initial audio sample. However, it can be seen that the decibel difference under low-pass filtering is very small. In the case of 8-bit reduction, the decibel difference decreases substantially because the number of bits in the modified samples is less than that in the initial audio sample. In Table 1, this decrease in the number of bits can be seen to considerably reduce the similarity rate for the original sample. In terms of the decibel range, except under the 8-bit reduction technique, the number of decibels was maintained between 70 and 90 dB; under 8-bit reduction the number of decibels was between 40 and 50 dB.

In terms of the difference in decibels, the table shown in Fig. 14 presents, for each technique, the difference between the number of decibels for the sample before the modification and that after the modification. The results for the initial audio sample show that

Original sentence: "this is for you"

Sentence recognized from Figure 8 (a): "this is for you"

Sentence recognized from Figure 8 (b): "example"

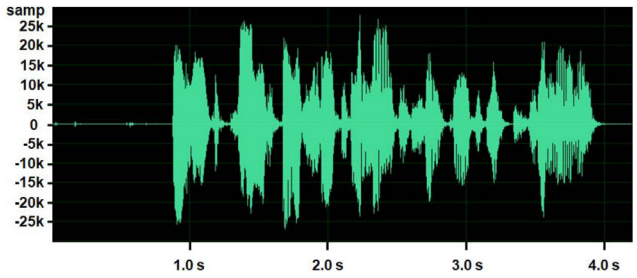
Sentence recognized from Figure 8 (c): "this is for youd"

Sentence recognized from Figure 8 (d): "tiso for youd"

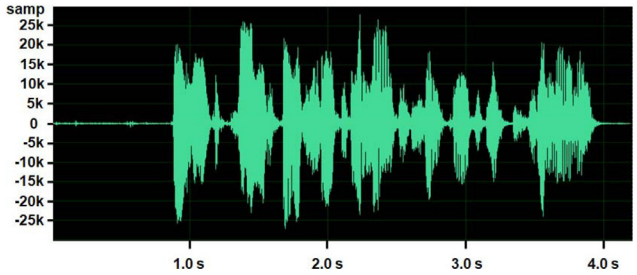
Fig. 10. Sentences recognized by DeepSpeech from waveforms shown in Fig. 8.

Table 1
Similarity rates for original samples and adversarial examples under each audio modification technique: low-pass filtering, 8-bit reduction, and audio silence removal. "Orig." is original sample; "adv." is adversarial example.

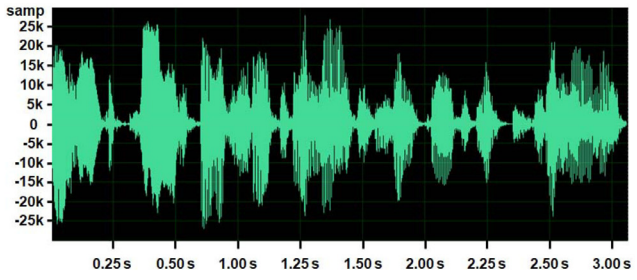
Parameter or metric	Low-pass filtering	8-bit reduction	Silence removal
Roll-off	12 dB	–	–
Similarity rate for orig.	93.94%	57.75%	71.44%
Similarity rate for adv.	6.21%	1.27%	0.66%



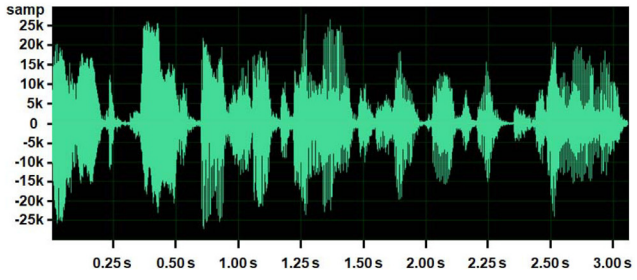
(a) Original sample (before)



(b) Adversarial example (before)



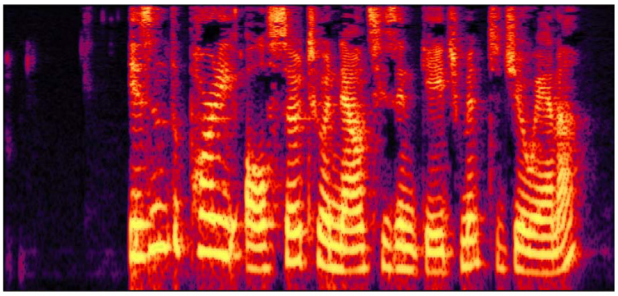
(c) Original sample (after)



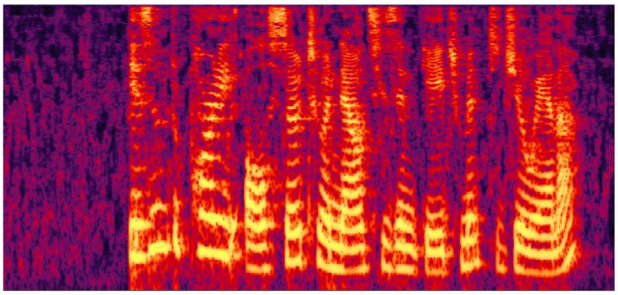
(d) Adversarial example (after)

Fig. 11. Waveforms for an original sample and an audio adversarial example, before and after audio silence removal. "Before" is before audio silence removal; "after" is after audio silence removal.

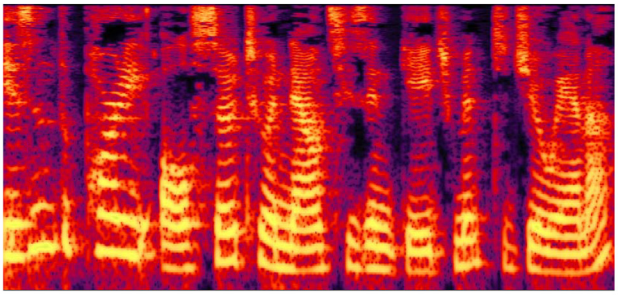
the decibel differences under the low-pass filtering and the audio silence removal techniques are about 2 dB and 0.12 dB, respectively, whereas under 8-bit reduction it is about 37 dB.



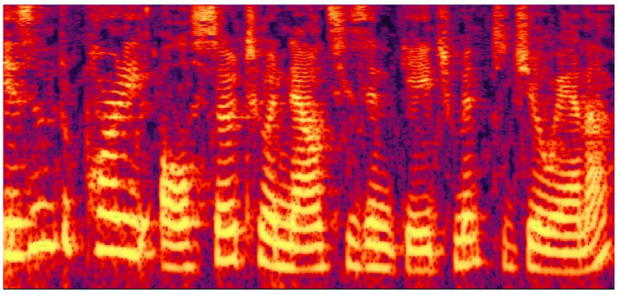
(a) Original sample (before)



(b) Adversarial example (before)



(c) Original sample (after)



(d) Adversarial example (after)

Fig. 12. Spectra for waveforms shown in Fig. 11. "Before" is before audio silence removal; "after" is after audio silence removal.

Fig. 15 shows the detection rates for adversarial examples and the error rates for original samples through audio modifications applied using the low-pass filtering (with 12 dB roll-off), 8-bit reduction, and silence removal techniques. If the input has a similarity rate lower than the threshold, it is determined to be an adversarial example, and if the input has a similarity rate higher than the threshold, it is determined to be an original sample. As seen in the figure, as the threshold increases, the detection rate for adversarial examples and the error rate for original samples increase. In terms of the detection rate for adversarial examples, the silence removal technique has better performance than the other techniques, achieved by effectively removing adversarial

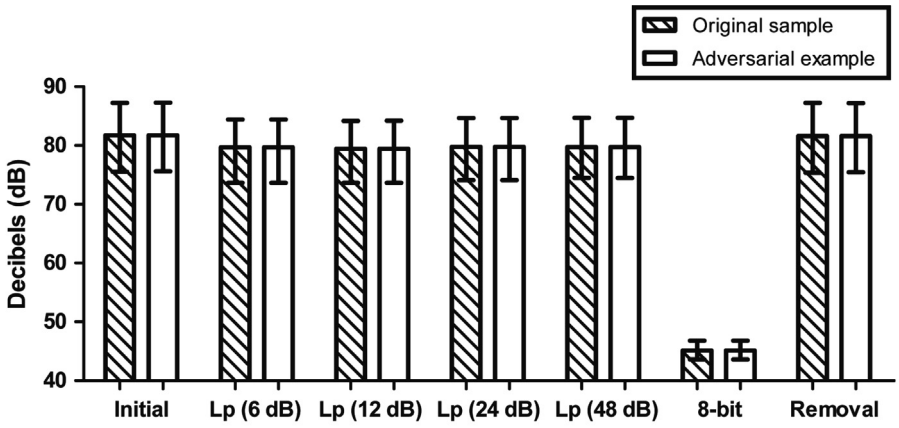
Original sentence: "isn't the party also to announce his engagement to joanna"

Sentence recognized from Figure 11 (a) and Figure 11 (c): "isn't the party also to announce his engagement to joanna"

Sentence recognized from Figure 11 (b): "example"

Sentence recognized from Figure 11 (d): "isot party also to announce his engagtment to joanna"

Fig. 13. Sentences recognized by DeepSpeech from waveforms shown in Fig. 11.



Metric	Low-pass filtering				8-bit	Removal
	6 dB	12 dB	24 dB	48 dB		
Difference for orig.	1.94	1.97	2.03	2.24	36.57	0.12
Difference for adv.	1.94	1.97	2.04	2.25	36.56	0.13

Fig. 14. The decibels and decibel differences for original samples and adversarial examples under each modification technique, using 100 test data. The lower and upper bounds of the bars are the standard 25th and 75th percentiles, respectively. "Initial" means initial audio sample before audio modification; "Lp" means low-pass filtering; "8-bit" means 8-bit reduction; "removal" means audio silence removal.

noise. In terms of the detection rate for original samples, the low-pass filtering technique, which does not significantly damage the original sample, has better performance than the other techniques. The sweet spot for the threshold value was 0.4: At this point, the silence removal and low-pass filtering techniques produced low error rates for original samples and high detection rates for adversarial examples.

5.2.5. Comparison with other defense methods

Table 2 shows the detection rates for adversarial examples and the error rates for original samples by the white noise method, denoise down-sampling method, temporal dependency method, and proposed method. The error rate is the proportion of times an original sample is incorrectly determined to be an adversarial example. The Mozilla common voice dataset and the DeepSpeech model were used as the dataset and target model, respectively.

With the white noise method, by which white noise is added to the input data at 20 dB, the detection rate for adversarial examples was 91%, and the error rate for original samples was 46%. The error rate for original samples was increased by adding white noise, which can affect the recognition of the original sample. With the denoise down-sampling method, the detection rate for adversarial examples was 92%, and the accuracy for original samples was 21%, using a frequency fluctuation of 8 kHz on the sound file. The DeepSpeech model, which is optimized for a sampling rate of 16 kHz, has poor accuracy for speech at sampling rates other than 16 kHz, and so the DeepSpeech model had an error rate of 21% for original samples. With the temporal dependency method, the detection rate for adversarial examples was 94%, and the error rate for original samples was 9%. In the process of segmenting the audio sample, endpoints are cut off or voice overlap occurs, and recognition of the original speech sample is partially lost; therefore, the temporal dependency method had an error rate of 9% for original samples. With the proposed method, when a low-pass filter with a roll-off of 12 dB was used for audio modification and the "sweet spot" threshold value of 0.4 was selected, the detection rate for adversarial examples was 97%, and the error rate for original samples was 5%. This method removes the high-frequency adversarial

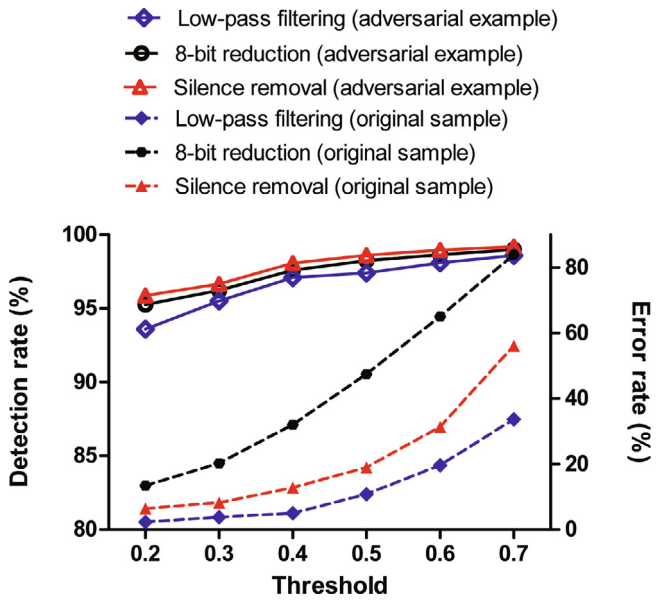


Fig. 15. Detection rates for adversarial examples and error rates for original samples through audio modifications applied using low-pass filtering (with 12 dB roll-off), 8-bit reduction, and silence removal techniques.

noise in the adversarial example while keeping the similarity rate for the original sample high, as the voice file provided contains few high-frequency voice bands.

6. Discussion

In this section, we discuss the proposed method as it relates to generating audio adversarial examples, audio modification techniques, applicability in ensemble methods, threshold value, and limitations.

6.1. Generation of audio adversarial examples

"Although the proposed scheme is designed as a defense method against audio adversarial examples, it can also be used to create an audio adversarial example regardless of its relation to the original sample and the length of the target phrase. This is because under the Carlini method [5], with its high attack success rate, there is no limit on the relationship to the original sample or the length of the target phrase, in contrast with other methods [16].

6.2. Audio modification technique

The proposed method performs detection by using audio modification, exploiting the fact that the adversarial example is more sensitive than the original sample. However, depending on the audio modification technique used, there may be too much distortion induced, causing the similarity rate for the original sample to be severely reduced, or, alternatively, there may be too little distortion and therefore no change in the adversarial example. For example, the filtering technique was tested using a high-pass filter, a notch filter, and other filter types, but these could not be used as audio modification techniques because they did not change the adversarial example. Therefore, the performance of the audio modification technique needs to be considered in advance.

In addition, the various audio modification techniques differ in performance, so it is necessary to consider their advantages and disadvantages. For example, audio silence removal produces a lower similarity rate than low-pass filtering on original samples, but also a lower similarity rate than low-pass filtering on adversarial examples. Thus, it is also necessary for the defender to consider this trade-off in selecting the most appropriate audio modification technique.

6.3. Justification of the proposed scheme

For the audio modification, we tested several techniques, including low-pass filtering, high-pass filtering, notch filtering, 8-bit reduction, hard-clipping, and silence removal. Of these, the low-pass filtering, 8-bit reduction, and audio silence removal techniques were found to be suitable for the audio modification. The use of these three techniques is justified by the fact that they can reduce the similarity rate for adversarial examples by removing or manipulating some adversarial noise while maintaining the similarity rate for original samples to some extent. The low-pass filtering technique removes the high-frequency range from audio sound. This technique reduces the similarity rate for adversarial examples by removing the high-frequency adversarial noise in the adversarial example while keeping the similarity rate for original samples high, as the voice file provided contains few high-frequency voice bands. The 8-bit reduction technique reduces 16-bit audio samples to 8-bit audio samples. While preserving the similarity rate for original samples to some extent, this technique reduces the similarity rate for adversarial examples by removing some of the adversarial noise in the adversarial example by reducing the dimensionality. The audio silence removal technique removes unnecessary silence from the sound. This technique reduces the similarity rate for adversarial examples by removing the adversarial noise of the silent region from the voice, while keeping the similarity rate for original samples slightly higher.

In audio modification, the similarity rate for original samples should be kept high, and the similarity rate for adversarial examples should be kept low. However, with the high-pass filtering, notch filtering, and hard-clipping techniques, a performance degradation occurred, in which the similarity rate for the original sample was remarkably low or the similarity rate for the adversarial example was high. The high-pass filtering technique removes low-frequency bands. When this technique was applied, the low-frequency band of the original sample was largely removed, causing the similarity rate for the original sample to drop below 21.3%. The notch filtering technique removes a specific frequency band. When this technique was applied, the specific frequency band was also removed in the original sample, causing the similarity rate for the original sample to drop to 32.6%. The hard-clipping technique is a type of distortion effect in which the amplitude of the signal is limited to a given maximum amplitude. When this

Table 2
Detection rates for adversarial examples and error rates for original samples by the white noise method, denoise down-sampling method, temporal dependency method, and proposed method, on the Mozilla common voice dataset and using the DeepSpeech model. "Orig." is original sample; "adv." is adversarial example.

Metric	White noise	Denoise down-sampling	Temporal dependency	Proposed
Detection rate for adv.	91%	92%	94%	97%
Error rate for orig.	46%	21%	9%	5%

technique was applied, little of the adversarial noise was removed, and the similarity rate for the adversarial example was high, 71.5%.

6.4. Applicability in ensemble methods

The proposed method can be used in combination with different audio modification techniques. For example, an adversarial example could be detected by using a combination of the low-pass filtering technique and the audio silence removal technique, as shown as type 1 and type 2 in Fig. 16. When two audio modification techniques are used, in this case low-pass filtering and audio silence removal, the proposed method compares the results before and after applying the two audio modifications. In calculating the similarity rate for original samples, because low-pass filtering is better, weight will be given to the original sample, whereas in calculating the similarity rate for adversarial examples, as audio silence removal is superior, weight will be given toward detection of the adversarial example. Combining techniques may enable the detection of audio adversarial examples with improved performance.

6.5. Similarity rate

By definition, the similarity rate is the proportion of matches between the initial classification result recognized before the audio modification and the classification result recognized after the audio modification. If there is no difference between the results of the classification before and after the audio modification, the similarity rate will be higher; if the difference between the results of classification before and after the audio modification is large, the similarity rate will be lower.

The low similarity rate for adversary examples shown in Table 1 is important because it means that the difference between the classification results before and after audio modification is large; with a low similarity rate, adversarial examples can be easily detected. An attacker creates an adversarial example by adding some noise to deceive the target classifier into classifying the input as the target classification, but after the input is passed through the audio modification, the attack success rate of the adversarial example decreases, and the adversarial example is recognized as the original classification.

In addition, the lower the similarity rate for adversarial examples, the higher the rate of detection of adversarial examples. The similarity rate for the original sample is high owing to the small change in the classification result from the audio modification, whereas an adversarial example has a large change in the classification result from the audio modification, resulting in a lower similarity rate for the adversarial example. The lower the similarity rate, the greater the gap with the similarity rate for the original sample, and the easier it is to detect adversarial examples.

6.6. Threshold

The proposed scheme is able to detect an adversarial example because of the difference in the results before and after the audio modification is applied. The experimental results show that for original samples, 93.94%, 57.75%, and 71.44% similarity rates are maintained with low-pass filtering, 8-bit reduction, and audio silence removal, respectively. For adversarial examples, on the other hand, the similarity rates are under 7%. Using the difference in similarity rates between original samples and adversarial examples, it was found that when the threshold is 0.4, the silence removal and low-pass filtering techniques produced low error rates for original samples and high detection rates for adversarial examples, with reference to Fig. 15. However, as each technique has different levels of average similarity rate for original samples and adversarial examples, it is necessary to select an appropriate threshold value for each technique.

6.7. Limitations

The proposed scheme can provide a limited-world defense for a speech recognition system. Our experiments have been with direct. wav files that do not include noise such as that from microphones, speakers, the room environment, and other noise sources that may be present for sounds transmitted through the air. If an adversarial example is affected by distortions caused by audio compression encoding, microphones, the indoor environment, the playback speaker, and other noise sources in an over-the-air transmission, the performance of the proposed scheme may be decreased. Therefore, future research will be expanded to include audio adversarial example defense methods that are effective for sounds transmitted through the air.

7. Conclusions

In this paper, we have proposed an acoustic-decoy method for detecting audio adversarial examples through audio modification.

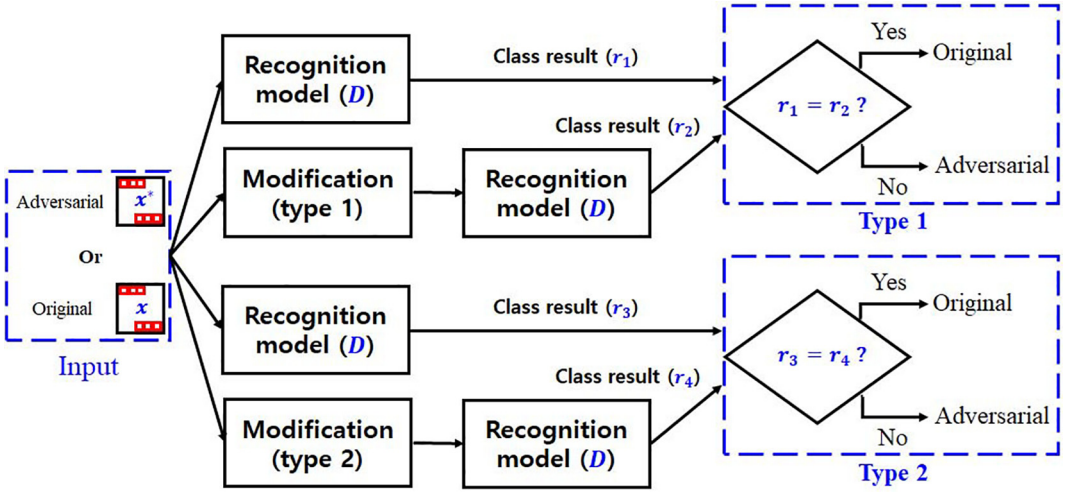


Fig. 16. Ensemble method using proposed method.

The key feature of the proposed scheme is the addition of well-formalized distortions using audio modification, enabling the classification result of an adversarial example to reflect changes because of its sensitivity, whereas an original sample will undergo only a slight change in its classification result. Experimental results show that the proposed method can detect adversarial examples by reducing the similarity rate for an adversarial example to 6.21%, 1.27%, and 0.66% using the low-pass filtering (with 12 dB roll-off), 8-bit reduction, and audio silence removal techniques, respectively. It can detect an audio adversarial example with a success rate of 97% by performing a comparison with the initial audio sample.

Future work will extend the method to other domains such as the video domain and for malware detection. Another challenge will be to develop a method for detecting an adversarial example generated in a real-world external environment. In addition, future research will examine the proposed defense method in the context of an ensemble strategy.

CRediT authorship contribution statement

Hyun Kwon: Conceptualization, Formal analysis, Methodology, Writing - original draft, Validation, Visualization. **Hyunsoo Yoon:** Supervision, Writing - review & editing, Investigation. **Ki-Woong Park:** Supervision, Writing - review & editing, Investigation, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) (NRF-2020R1A2C4002737).

Appendix A

The following are fifteen examples of Mozilla voice test data [31]. (1) that day the merchant gave the boy permission to build the display; (2) he was going to miss the place and all the good things he had learned; (3) it was dropping off in flakes and raining down on the sand; (4) the shower's in there; (5) follow the instructions here; (6) he remembered something his grandfather had once told him that butterflies were a good omen; (7) the shop is closed on mondays; (8) even coming down on the train together she wrote me; (9) i'm going away he said; (10) it must have fallen while i was sitting over there; (11) a huge hole had been made by the impact of the projectile; (12) it's candice now on long distance from washington; (13) he could always go back to being a shepherd; (14) the boy went to his room and packed his belongings; (15) their faces were hidden behind blue veils with only their eyes showing.

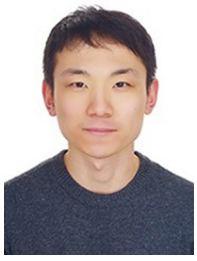
Table 3
Characteristics of Mozilla voice test data [31], including average duration and text length. "Chars" is characters.

Description	Value
Quantity	100
Average duration	6.023 s
Average text length	44.62 chars
Sampling rate	16,000 Hz
Number of sampling bits	16 bits

References

[1] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117.
[2] T. Vaidya, Y. Zhang, M. Sherr, C. Shields, Cocaine noodles: exploiting the gap between human and machine speech recognition, WOOT 15 (2015) 10–11.
[3] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, W. Zhou, Hidden voice commands., in: USENIX Security Symposium, 2016, pp. 513–530.
[4] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, W. Xu, Dolphinattack: Inaudible voice commands, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2017, pp. 103–117.
[5] N. Carlini, D. Wagner, Audio adversarial examples: Targeted attacks on speech-to-text, Deep Learning and Security Workshop (2018).
[6] B. Logan, et al., Mel frequency cepstral coefficients for music modeling., in: ISMIR, vol. 270, 2000, pp. 1–11.
[7] 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 preprint arXiv:1412.5567 (2014).
[8] H. Kwon, H. Yoon, K.-W. Park, Poster: Detecting audio adversarial example through audio modification, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19, ACM, 2019, pp. 2521–2523.
[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.
[10] S.R. Eddy, Hidden markov models, Current Opinion in Structural Biology 6 (3) (1996) 361–365.
[11] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 369–376.
[12] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, in: Eleventh Annual Conference of the International Speech Communication Association, 2010.
[13] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: Security and Privacy (SP), 2017 IEEE Symposium on, IEEE, 2017, pp. 39–57.
[14] https://goo.gl/5yANXA.
[15] G.L. Oliveira, A. Valada, C. Bollen, W. Burgard, T. Brox, Deep learning for human part discovery in images, in: Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, 2016, pp. 1634–1641.
[16] M. Cisse, Y. Adi, N. Neverova, J. Keshet, Houdini: Fooling deep structured prediction models, arXiv preprint arXiv:1707.05373 (2017).
[17] M. Alzantot, B. Balaji, M. Srivastava, Did you hear that? adversarial examples against automatic speech recognition, arXiv preprint arXiv:1801.00554 (2018).
[18] H. Yu, Z.-H. Tan, Z. Ma, R. Martin, J. Guo, Spoofing detection in automatic speaker verification systems using dnn classifiers and dynamic acoustic features, IEEE Transactions on Neural Networks and Learning Systems 29 (10) (2017) 4633–4644.
[19] T. Fernando, S. Sridharan, M.L. McLaren, D. Priyasad, S. Denman, C. Fookes, Temporarily-aware context modelling using generative adversarial networks for speech activity detection, IEEE/ACM Transactions on Audio, Speech, and Language Processing (2020).
[20] V. Subramanian, E. Benetos, M.B. Sandler, Robustness of adversarial attacks in sound event classification (2019).
[21] K. Tamura, A. Omagari, S. Hashida, Novel defense method against audio adversarial example for speech-to-text transcription neural networks, in: 2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA), IEEE, 2019, pp. 115–120.
[22] Z. Yang, B. Li, P.-Y. Chen, D. Song, Characterizing audio adversarial examples using temporal dependency, arXiv preprint arXiv:1809.10875 (2018).
[23] E.V. Raghavendra, P. Vijayaditya, K. Prahallad, Speech synthesis using artificial neural networks, in: Communications (NCC), 2010 National Conference on, IEEE, 2010, pp. 1–5.
[24] R.B. Himmelstein, Voice-controlled vehicle control system, uS Patent 6,496,107 (Dec. 17 2002).
[25] F.-Y. Leu, G.-L. Lin, An mfcc-based speaker identification system, in: Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on, IEEE, 2017, pp. 1055–1062.
[26] L. Palen, M. Salzman, Voice-mail diary studies for naturalistic data capture under mobile conditions, in: Proceedings of the 2002 ACM conference on Computer supported cooperative work, ACM, 2002, pp. 87–95.
[27] F. Wang, D. Tian, Y. Wang, High accuracy inertial stabilization via kalman filter based disturbance observer, in: 2016 IEEE International Conference on Mechatronics and Automation, IEEE, 2016, pp. 794–802.
[28] T.B. Aji, J. Raharjo, L. Novamizanti, Analisis audio watermarking berbasis dwt dengan metode qr decomposition dan quantization index menggunakan pso: Analysis audio watermarking based on dwt using qr decomposition and quatization index use pso, eProceedings of Engineering 6 (1) (2019).
[29] M. Mojiri, A.R. Bakhshai, An adaptive notch filter for frequency estimation of a periodic signal, IEEE Transactions on Automatic Control 49 (2) (2004) 314–318.
[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning., in: OSDI, vol. 16, 2016, pp. 265–283.

[31] https://voice.mozilla.org/.
[32] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: The International Conference on Learning Representations (ICLR), 2015.
[33] https://bit.ly/2rfhdwD.
[34] https://bit.ly/2NpZ0Fr.
[35] https://bit.ly/2K0M1I5.
[36] https://bit.ly/2oUVGbO.
[37] https://bit.ly/2WRK5XD.
[38] https://bit.ly/32qWram.
[39] https://bit.ly/36LXSDQ.
[40] https://bit.ly/33BexrL.
[41] J.T. Barnett, B. Kedem, Zero-crossing rates of functions of gaussian processes, IEEE Transactions on Information Theory 37 (4) (1991) 1188–1194.
[42] M. Jalil, F.A. Butt, A. Malik, Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals, in: 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), IEEE, 2013, pp. 208–212.
[43] http://bit.ly/34BS3qu.
[44] http://bit.ly/2WWCYNx.
[45] http://bit.ly/2WUdgcM.
[46] http://bit.ly/33rdvOH.



Hyun Kwon received the B.S degree in mathematics from Korea Military Academdy, South Korea, in 2010. He also received the M.S. degree in School of Computing from Korea Advanced Institute of Science and Technology (KAIST) in 2015, and the Ph.D. degree at School of Computing, KAIST in 2020. He is currently an assistant professor in Korea Military Academy. His research interests include information security, machine learning, computer security, and intrusion tolerant system.

Hyunsoo Yoon received the B.E. degree in electronic-sengineering from Seoul National University, South Korea, in 1979, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1981, and the Ph.D. degree in computer and information science from the Ohio State University, Columbus, Ohio, in 1988. From 1988 to 1989, he was a member of technical staff at AT&T Bell Labs. Since 1989 he has been a faculty member of School of Computing at KAIST. His main research interest includes wireless sensor networks, 4G networks, and network security.

Ki-Woong Park received the B.S. degree in computer science from Yonsei University, South Korea, in 2005, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007, and the Ph.D. degree in electrical engineering from KAIST in 2012. He received a 2009–2010 Microsoft Graduate Research Fellowship. He worked for National Security Research Institute as a senior researcher. He has been a professor in the department of computer and information security at Sejong University. His research interests include security issues for cloud and mobile computing systems as well as the actual system implementation and subsequent evaluation in a real computing system.