Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding

Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa
Ruhr University Bochum, Germany
{lea.schoenherr, katharina.kohls, steffen.zeiler, thorsten.holz, dorothea.kolossa}@rub.de

***Abstract*—Voice interfaces are becoming accepted widely as input methods for a diverse set of devices. This development is driven by rapid improvements in automatic speech recognition (ASR), which now performs on par with human listening in many tasks. These improvements base on an ongoing evolution of deep neural networks (DNNs) as the computational core of ASR. However, recent research results show that DNNs are vulnerable to adversarial perturbations, which allow attackers to force the transcription into a malicious output. In this paper, we introduce a new type of adversarial examples based on *psychoacoustic hiding*. Our attack exploits the characteristics of DNN-based ASR systems, where we extend the original analysis procedure by an additional backpropagation step. We use this backpropagation to learn the degrees of freedom for the adversarial perturbation of the input signal, i.e., we apply a psychoacoustic model and manipulate the acoustic signal below the thresholds of human perception. To further minimize the perceptibility of the perturbations, we use forced alignment to find the best fitting temporal alignment between the original audio sample and the malicious target transcription. These extensions allow us to embed an arbitrary audio input with a malicious voice command that is then transcribed by the ASR system, with the audio signal remaining barely distinguishable from the original signal. In an experimental evaluation, we attack the state-of-the-art speech recognition system *Kaldi* and determine the best performing parameter and analysis setup for different types of input. Our results show that we are successful in up to $98\,\%$ of cases with a computational effort of fewer than two minutes for a ten-second audio file. Based on user studies, we found that none of our target transcriptions were audible to human listeners, who still understand the original speech content with unchanged accuracy.**

I. Introduction

> Hello darkness, my old friend. I've come to talk with you again. Because a vision softly creeping left its seeds while I was sleeping. And the vision that was planted in my brain still remains, within the sound of silence.
>
> Simon & Garfunkel, *The Sound of Silence*

Motivation. Deep neural networks (DNNs) have evolved into the state-of-the-art approach for many machine learning tasks, including automatic speech recognition (ASR) systems [43], [57]. The recent success of DNN-based ASR systems is due to a number of factors, most importantly their power to model large vocabularies and their ability to perform speaker-independent and also highly robust speech recognition. As a result, they can cope with complex, real-world environments that are typical for many speech interaction scenarios such as voice interfaces. In practice, the importance of DNN-based ASR systems is steadily increasing, e. g., within smartphones or stand-alone devices such as Amazon's Echo/Alexa.

On the downside, their success also comes at a price: the number of necessary parameters is significantly larger than that of the previous state-of-the-art Gaussian-Mixture-Model probability densities within Hidden Markov Models (so-called GMM-HMM systems) [39]. As a consequence, this high number of parameters gives an adversary much space to explore (and potentially exploit) blind spots that enable her to mislead an ASR system. Possible attack scenarios include unseen requests to ASR assistant systems, which may reveal private information. Diao et al. demonstrated that such attacks are feasible with the help of a malicious app on a smartphone [14]. Attacks over radio or TV, which could affect a large number of victims, are another attack scenarios. This could lead to unwanted online shopping orders, which has already happened on normally uttered commands over TV commercials, as Amazon's devices have reacted to the purchase command [30]. As ASR systems are also often included into smart home setups, this may lead to a significant vulnerability and in a worst-case scenario, an attacker may be able to take over the entire smart home system, including security cameras or alarm systems.

Adversarial Examples. The general question if ML-based systems can be secure has been investigated in the past [5], [6], [26] and some works have helped to elucidate the phenomenon of adversarial examples [16], [18], [25], [47]. Much recent work on this topic focussed on image classification: different types of adversarial examples have been investigated [9], [15], [32] and in response, several types of countermeasures have been proposed [12], [19], [60]. These countermeasures are focused on only classification-based recognition and some approaches remain resistant [9]. As the recognition of ASR systems operates differently due to time dependencies, such countermeasures will not work equally in the audio domain.

In the audio domain, Vaidya et al. were among the first to explore adversarial examples against ASR systems [52]. They showed how an input signal (i. e., audio file) can be modified to fit the target transcription by considering the features instead of the output of the DNN. On the downside, the results show high distortions of the audio signal and a human can easily

Network and Distributed Systems Security (NDSS) Symposium 2019
24-27 February 2019, San Diego, CA, USA
ISBN 1-891562-55-X
https://dx.doi.org/10.14722/ndss.2019.23288
www.ndss-symposium.org

perceive the attack. Carlini et al. introduced so-called *hidden voice commands* and demonstrated that targeted attacks against HMM-only ASR systems are feasible [8]. They use inverse feature extraction to create adversarial audio samples. Still, the resulting audio samples are not intelligible by humans (in most of the cases) and may be considered as noise, but may make thoughtful listeners suspicious. To overcome this limitation, Zhang et al. proposed so-called *DolphinAttacks*: they showed that it is possible to hide a transcription by utilizing non-linearities of microphones to modulate the baseband audio signal with ultrasound higher than 20 kHz [61]. This work has considered ultrasound only, however, our psychoacoustics-based approach instead focuses on the human-perceptible frequency range. The drawback of this and similar ultrasound-based attacks [42], [48] is that the attack is costly as the information to manipulate the input features needs to be retrieved from recordings of audio signals with the specific microphone, which is used for the attack. Additionally, the modulation is tailored to a specific microphone, such that the result may differ if another microphone is used. Recently, Carlini and Wagner published a technical report in which they introduce a general targeted attack on ASR systems using connectionist temporal classification (CTC) loss [10]. Similarly to previous adversarial attacks on image classifiers, it works with a gradient-descent-based minimization [9], but it replaces the loss function by the CTC-loss, which is optimized for time sequences. On the downside, the constraint for the minimization of the difference between original and adversarial sample is also borrowed from adversarial attacks on images and therefore does not consider the limits and sensitivities of human auditory perception. Additionally, the algorithm often does not converge. This is solved by multiple initializations of the algorithm, which leads to high run-time requirements—in the order of hours of computing time—to calculate an adversarial example. Also recently, Yuan et al. described *CommanderSong*, which is able to hide transcripts within music [59]. However, this approach is only shown to be successful in music and it does not contain a human-perception-based noise reduction.

Contributions. In this paper, we introduce a novel type of adversarial examples against ASR systems based on *psychoacoustic hiding*. We utilize psychoacoustic modeling, as in MP3 encoding, in order to reduce the perceptible noise. For this purpose, hearing thresholds are calculated based on psychoacoustic experiments by Zwicker et al. [62]. This limits the adversarial perturbations to those parts of the original audio sample, where they are not (or hardly) perceptible by a human. Furthermore, we use backpropagation as one part of the algorithm to find adversarial examples with minimal perturbations. This algorithm has already been successfully used for adversarial examples in other settings [9], [10]. To show the general feasibility of psychoacoustic attacks, we feed the audio signal directly into the recognizer.

A key feature of our approach is the integration of the preprocessing step into the backpropagation. As a result, it is possible to change the raw audio signal without further steps. The preprocessing operates as a feature extraction and is fundamental to the accuracy of an ASR system. Due to the differentiability of each single preprocessing step, we are able to include it in the backpropagation without the necessity to invert the feature extraction. In addition, ASR highly depends on temporal alignment as it is a continuous process. We enhance our attack by computing an optimal alignment with the *forced alignment* algorithm, which calculates the best starting point for the backpropagation. Hence, we make sure to move the target transcription into parts of the original audio sample which are the most promising to not be perceivable by a human. We optimize the algorithm to provide a high success rate and to minimize the perceptible noise.

We have implemented the proposed attack to demonstrate the practical feasibility of our approach. We evaluated it against the state-of-the-art DNN-HMM-based ASR system *Kaldi* [38], which is one of the most popular toolchains for ASR among researchers [17], [27], [28], [40], [41], [50], [51], [53], [59] and is also used in commercial products such as Amazon's Echo/Alexa and by IBM and Microsoft [3], [58]. Note that commercial ASR systems do not provide information about their system setup and configuration.

Such information could be extracted via model stealing and similar attacks (e. g., [20], [34], [37], [49], [54]). However, such an end-to-end attack would go beyond the contributions of this work and hence we focus on the general feasibility of adversarial attacks on state-of-the-art ASR systems in a white-box setting. More specifically, we show that it is possible to hide any target transcription in any audio file with a minimum of perceptible noise in up to $98\,\%$ of cases. We analyze the optimal parameter settings, including different phone rates, allowed deviations from the hearing thresholds, and the number of iterations for the backpropagation. We need less than two minutes on an Intel Core i7 processor to generate an adversarial example for a ten-second audio file. We also demonstrate that it is possible to limit the perturbations to parts of the original audio files, where they are not (or only barely) perceptible by humans. The experiments show that in comparison to other targeted attacks [59], the amount of noise is significantly reduced.

This observation is confirmed during a two-part audibility study, where test listeners transcribe adversarial examples and rate the quality of different settings. The results of the first user study indicate that it is impossible to comprehend the target transcription of adversarial perturbations and only the original transcription is recognized by human listeners. The second part of the listening test is a MUSHRA test [44] in order to rate the quality of different algorithm setups. The results show that the psychoacoustic model greatly increases the quality of the adversarial examples.

In summary, we make the following contributions in this paper:

- **Psychoacoustic Hiding.** We describe a novel type of adversarial examples against DNN-HMM-based ASR systems based on a psychoacoustically designed attack for hiding transcriptions in arbitrary audio files. Besides the psychoacoustic modeling, the algorithm utilizes an optimal temporal alignment and backpropagation up to the raw audio file.
- **Experimental Evaluation.** We evaluate the proposed attack algorithm in different settings in order to find adversarial perturbations that lead to the best recognition result with the least human-perceptible noise.

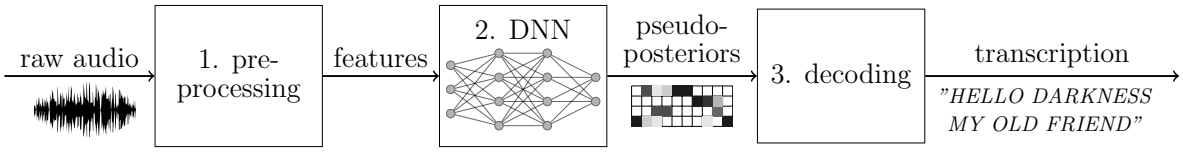


Fig. 1: Overview of a state-of-the-art ASR system with the three main components of the ASR system: (1) preprocessing of the raw audio data, (2) calculating pseudo-posteriors with a DNN, and (3) the decoding, which returns the transcription.

- **User Study.** To measure the human perception of adversarial audio samples, we performed a user study. More specifically, human listeners were asked to transcribe what they understood when presented with adversarial examples and to compare their overall audio quality compared to original unmodified audio files.

A demonstration of our attack is available online at https://adversarial-attacks.net where we present several adversarial audio files generated for different kinds of attack scenarios.

II. Technical Background

Neural networks have become prevalent in many machine learning tasks, including modern ASR systems. Formally speaking, they are just functions $y = F(x)$, mapping some input x to its corresponding output y. Training these networks requires the adaptation of hundreds of thousands of free parameters. The option to train such models by just presenting input-output pairs during the training process makes deep neural networks (DNNs) so appealing for many researchers. At the same time, this represents the Achilles' heel of these systems that we are going to exploit for our ASR attack. In the following, we provide the technical background as far as it is necessary to understand the details of our approach.

A. Speech Recognition Systems

There is a variety of commercial and non-commercial ASR systems available. In the research community, *Kaldi* [38] is very popular given that it is an open-source toolkit which provides a wide range of state-of-the-art algorithms for ASR. The tool was developed at Johns Hopkins University and is written in C++. We performed a partial reverse engineering of the firmware of an Amazon Echo and our results indicate that this device also uses *Kaldi* internally to process audio inputs. Given *Kaldi*'s popularity and its accessibility, this ASR system hence represents an optimal fit for our experiments. Figure 1 provides an overview of the main system components that we are going to describe in more detail below.

1) Preprocessing Audio Input: Preprocessing of the audio input is a synonym for feature extraction: this step transforms the raw input data into features that should ideally preserve all relevant information (e. g., phonetic class information, formant structure, etc.), while discarding the unnecessary remainder (e. g., properties of the room impulse response, residual noise, or voice properties like pitch information). For the feature extraction in this paper, we divide the input waveform into overlapping frames of fixed length. Each frame is transformed individually using the discrete Fourier transform (DFT) to obtain a frequency domain representation. We calculate the logarithm of the magnitude spectrum, a very common feature representation for ASR systems. A detailed description is given in Section III-E, where we explain the necessary integration of this particular preprocessing into our ASR system.

2) Neural Network: Like many statistical models, an artificial neural network can learn very general input/output mappings from training data. For this purpose, so-called neurons are arranged in layers and these layers are stacked on top of each other and are connected by weighted edges to form a DNN. Their parameters, i. e., the weights, are adapted during the training of the network. In the context of ASR, DNNs can be used differently. The most attractive and most difficult application would be the direct transformation of the spoken text at the input to a character transcription of the same text at the output. This is referred to as an end-to-end-system. *Kaldi* takes a different route: it uses a more conventional Hidden Markov Model (HMM) representation in the decoding stage and uses the DNN to model the probability of all HMM states (modeling context-dependent phonetic units) given the acoustic input signal. Therefore, the outputs of the DNN are pseudo-posteriors, which are used during the decoding step in order to find the most likely word sequence.

3) Decoding: Decoding in ASR systems, in general, utilizes some form of graph search for the inference of the most probable word sequence from the acoustic signal. In *Kaldi*, a static decoding graph is constructed as a composition of individual transducers (i. e., graphs with input/output symbol mappings attached to the edges). These individual transducers describe for example the grammar, the lexicon, context dependency of context-dependent phonetic units, and the transition and output probability functions of these phonetic units. The transducers and the pseudo-posteriors (i. e., the output of the DNN) are then used to find an optimal path through the word graph.

B. Adversarial Machine Learning

Adversarial attacks can, in general, be applied to any kind of machine learning system [5], [6], [26], but they are successful especially for DNNs [18], [35].

As noted above, a trained DNN maps an input x to an output $y = F(x)$. In the case of a trained ASR system, this is a mapping of the features into estimated pseudo-posteriors. Unfortunately, this mapping is not well defined in all cases due to the high number of parameters in the DNN, which leads to a very complex function $F(x)$. Insufficient generalization of $F(x)$ can lead to blind spots, which may not be obvious to humans. We exploit this weakness by using a manipulated input x' that closely resembles the original input x, but leads to a different mapping:

$$x' = x + \delta, \quad \text{such that } F(x) \neq F(x'),$$

where we minimize any additional noise δ such that it stays close to the hearing threshold. For the minimization, we use a model of human audio signal perception. This is easy for cases where no specific target y' is defined. In the following, we show that adversarial examples can even be created very reliably for targeted attacks, where the output y' is defined.

C. Backpropagation

Backpropagation is an optimization algorithm for computational graphs (like those of neural networks) based on gradient descent. It is normally used during the training of DNNs to learn the optimal weights. With only minor changes, it is possible to use the same algorithm to create adversarial examples from arbitrary inputs. For this purpose, the parameters of the DNN are kept unchanged and only the input vector is updated.

For backpropagation, three components are necessary:

1) **Measure loss.** The difference between the actual output $y_i = F(x_i)$ and the target output y' is measured with a loss function $L(y_i, y')$. The index i denotes the current iteration step, as backpropagation is an iterative algorithm. The cross-entropy, a commonly used loss function for DNNs with classification tasks, is employed here S

$$L(y_i, y') = -\sum y_i \log(y').$$

2) **Calculate gradient.** The loss is back-propagated to the input x_i of the neural network. For this purpose, the gradient ∇x_i is calculated by partial derivatives and the chain rule

$$\nabla x_i = \frac{\partial L(y_i, y')}{\partial x_i} = \frac{\partial L(y_i, y')}{\partial F(x_i)} \cdot \frac{\partial F(x_i)}{\partial x_i}. \quad (1)$$

The derivative of $F(x_i)$ depends on the topology of the neural network and is also calculated via the chain rule, going backward through the different layers.

3) **Update.** The input is updated according to the back-propagated gradient and a learning rate α via

$$x_{i+1} = x_i - \nabla x_i \cdot \alpha.$$

These steps are repeated until convergence or until an upper limit for the number of iterations is reached. With this algorithm, it is possible to approximately solve problems iteratively, which cannot be solved analytically. Backpropagation is guaranteed to find a minimum, but not necessarily the global minimum. As there is not only one solution for a specific target transcription, it is sufficient for us to find *any* solution for a valid adversarial example.

D. Psychoacoustic Modeling

Psychoacoustic hearing thresholds describe how the dependencies between frequencies lead to masking effects in the human perception. Probably the best-known example for this is MP3 compression [21], where the compression algorithm applies a set of empirical hearing thresholds to the input signal. By removing those parts of the input signal that are inaudible by human perception, the original input signal can be transformed into a smaller but lossy representation.

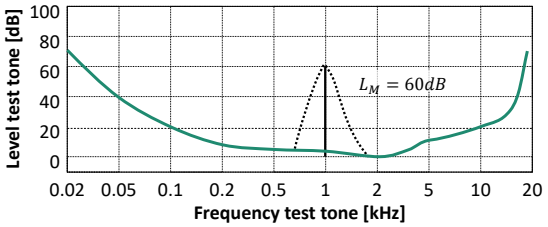


Fig. 2: Hearing threshold of test tone (dashed line) masked by a $L_{CB} = 60dB$ tone at 1 kHz [62]. In green, the hearing threshold in quiet is shown.

1) Hearing Thresholds: MP3 compression depends on an empirical set of hearing thresholds that define how dependencies between certain frequencies can mask, i. e., make inaudible, other parts of an audio signal. The thresholds derived from the audio do not depend on the audio type, e.g., whether music or speech was used. When applied to the frequency domain representation of an input signal, the thresholds indicate which parts of the signal can be altered in the following quantization step, and hence, help to compress the input. We utilize this psychoacoustic model for our manipulations of the signal, i. e., we apply it as a rule set to *add* inaudible noise. We derive the respective set of thresholds for an audio input from the psychoacoustic model of MP3 compression. In Figure 2 an example for a single tone masker is shown. Here, the green line represents the human hearing thresholds in quiet over the complete human-perceptible frequency range. In case of a masking tone, this threshold increases, reflecting the decrease in sensitivity in the frequencies around the test tone. In Figure 2 this is shown for $1\,\mathrm{kHz}$ and $60\,\mathrm{dB}$.

2) MP3 Compression: We receive the original input data in buffers of 1024 samples length that consist of two 576 sample *granule* windows. One of these windows is the current granule, the other is the previous granule that we use for comparison. We use the fast Fourier transform to derive 32 frequency bands from both granules and break this spectrum into MPEG ISO [21] specified scale factor bands. This segmentation of frequency bands helps to analyze the input signal according to its acoustic characteristics, as the hearing thresholds and masking effects directly relate to the individual bands. We measure this segmentation of bands in *bark*, a subjective measurement of frequency. Using this bark scale, we estimate the *relevance* of each band and compute its energy.

In the following steps of the MP3 compression, the thresholds for each band indicate which parts of the frequency domain can be removed while maintaining a certain audio quality during quantization. In the context of our work, we use the hearing thresholds as a guideline for acceptable manipulations of the input signal. They describe the amount of energy that can be added to the input in each individual window of the signal. An example of such a matrix is visualized in Figure 5d. The matrices are always normalized in such a way that the largest time-frequency-bin energy is limited to $95\,\mathrm{dB}$.

III. Attacking ASR via Psychoacoustic Hiding

In the following, we show how the audible noise can be limited by applying hearing thresholds during the creation of

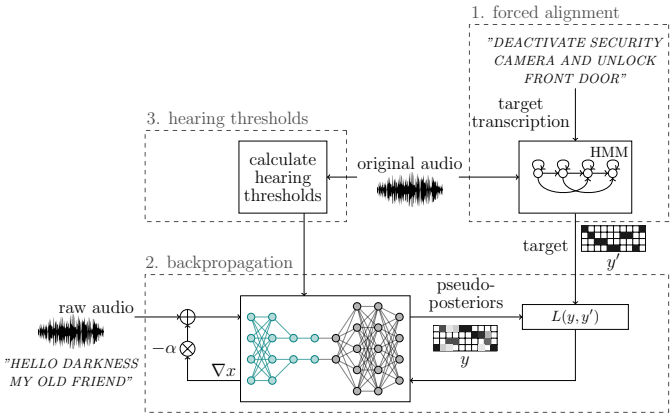


Fig. 3: The creation of adversarial examples can be divided into three components: (1) *forced alignment* to find an optimal target for the (2) backpropagation and the integration of (3) the hearing thresholds.

adversarial examples. As an additional challenge, we need to find the optimal temporal alignment, which gives us the best starting point for the insertion of malicious perturbations. Note that our attack integrates well into the DNN-based speech recognition process: we use the trained ASR system and apply backpropagation to update the input, eventually resulting in adversarial examples. A demonstration of our attack is available at https://adversarial-attacks.net.

A. Adversary Model

Throughout the rest of this paper, we assume the following adversary model. First, we assume a white-box attack, where the adversary knows the ASR mechanism of the attacked system. Using this knowledge, the attacker generates audio samples containing malicious perturbations before the actual attack takes place, i. e., the attacker exploits the ASR system to obtain an audio file that produces the desired recognition result. Second, we assume the ASR system to be configured in such a way that it gives the best possible recognition rate. In addition, the trained ASR system, including the DNN, remains unchanged over time. Finally, we assume a perfect transmission channel for replaying the manipulated audio samples, hence, we do not take perturbations through audio codecs, compression, hardware, etc. into account by feeding the audio file directly into the recognizer. Note that we only consider targeted attacks, where the target transcription is predefined (i. e., the adversary chooses the target sentence).

B. High-Level Overview

The algorithm for the calculation of adversarial examples can be divided into three parts, which are sketched in Figure 3. The main difference between *original audio* and *raw audio* is that the *original audio* does not change during the run-time of the algorithm, but the *raw audio* is updated iteratively in order to result in an adversarial example. Before the backpropagation, the best possible temporal alignment is calculated via so-called *forced alignment*. The algorithm uses the original audio signal and the target transcription as inputs in order to find the best target pseudo-posteriors. The *forced alignment* is performed once at the beginning of the algorithm.

With the resulting target, we are able to apply backpropagation to manipulate our input signal in such a way that the speech recognition system transcribes the desired output. The backpropagation is an iterative process and will, therefore, be repeated until it converges or a fixed upper limit for the number of iterations is reached.

The hearing thresholds are applied during the backpropagation in order to limit the changes that are perceptible by a human. The hearing thresholds are also calculated once and stored for the backpropagation. A detailed description of the integration is provided in Section III-F.

C. Forced Alignment

One major problem of attacks against ASR systems is that they require the recognition to pass through a certain sequence of HMM states in such a way that it leads to the target transcription. However, due to the decoding step—which includes a graph search—for a given transcription, many valid pseudo-posterior combinations exist. For example, when the same text is spoken at different speeds, the sequence of the HMM states is correspondingly faster or slower. We can benefit from this fact by using that version of pseudo-posteriors which best fits the given audio signal and the desired target transcription.

We use forced alignment as an algorithm for finding the best possible temporal alignment between the acoustic signal that we manipulate and the transcription that we wish to obtain. This algorithm is provided by the *Kaldi* toolkit. Note that it is not always possible to find an alignment that fits an audio file to any target transcription. In this case, we set the alignment by dividing the audio sample equally into the number of states and set the target according to this division.

D. Integrating Preprocessing

We integrate the preprocessing step and the DNN step into one joint DNN. This approach is sketched in Figure 4. The input for the preprocessing is the same as in Figure 1, and the pseudo-posteriors are also unchanged. For presentation purposes, this is only a sketch of the DNN, the used DNN contains far more neurons.

This design choice does not affect the accuracy of the ASR system, but it allows for manipulating the raw audio data by applying backpropagation to the preprocessing steps, directly giving us the optimally adversarial audio signal as result.

E. Backpropagation

Due to this integration of preprocessing into the DNN, Equation (1) has to be extended to

$$\nabla x = \frac{\partial L(y, y')}{\partial F(\chi)} \cdot \frac{\partial F(\chi)}{\partial F_P(x)} \cdot \frac{\partial F_P(x)}{\partial x},$$

where we ignore the iteration index i for simplicity. All preprocessing steps are included in $\chi = F_P(x)$ and return the input features χ for the DNN. In order to calculate $\frac{\partial F_P(x)}{\partial x}$, it is necessary to know the derivatives of each of the four preprocessing steps. We will introduce these preprocessing steps and the corresponding derivatives in the following.

1) Framing and Window Function: In the first step, the raw audio data is divided into T frames of length N and a window function is applied to each frame. A window function is a simple, element-wise multiplication with fixed values $w(n)$

$$x_w(t,n) = x(t,n) \cdot w(n), \quad n = 0, \dots, N-1,$$

with $t = 0, \dots, T-1$. Thus, the derivative is just

$$\frac{\partial x_w(t,n)}{\partial x(t,n)} = w(n).$$

2) Discrete Fourier Transform: For transforming the audio signal into the frequency domain, we apply a DFT to each frame x_w. This transformation is a common choice for audio features. The DFT is defined as

$$X(t,k) = \sum_{n=0}^{N-1} x_w(t,n) e^{-i2\pi \frac{kn}{N}}, \quad k = 0, \dots, N-1.$$

Since the DFT is a weighted sum with fixed coefficients $e^{-i2\pi \frac{kn}{N}}$, the derivative for the backpropagation is simply the corresponding coefficient

$$\frac{\partial X(t,k)}{\partial x_w(t,n)} = e^{-i2\pi \frac{kn}{N}}, \quad k, n = 0, \dots, N-1.$$

3) Magnitude: The output of the DFT is complex valued, but as the phase is not relevant for speech recognition, we just use the magnitude of the spectrum, which is defined as

$$|X(t,k)|^2 = \mathrm{Re}(X(t,k))^2 + \mathrm{Im}(X(t,k))^2,$$

with $\mathrm{Re}(X(t,k))$ and $\mathrm{Im}(X(t,k))$ as the real and imaginary part of $X(t,k)$. For the backpropagation, we need the derivative of the magnitude. In general, this is not well defined and allows two solutions,

$$\frac{\partial |X(t,k)|^2}{\partial X(t,k)} = \begin{cases} 2 \cdot \mathrm{Re}(X(t,k)) \\ 2 \cdot \mathrm{Im}(X(t,k)) \end{cases}.$$

We circumvent this problem by considering the real and imaginary parts separately and calculate the derivatives for both cases

$$\nabla X(t,k) = \begin{pmatrix} \frac{\partial |X(t,k)|^2}{\partial \mathrm{Re}(X(t,k))} \\ \frac{\partial |X(t,k)|^2}{\partial \mathrm{Im}(X(t,k))} \end{pmatrix} = \begin{pmatrix} 2 \cdot \mathrm{Re}(X(t,k)) \\ 2 \cdot \mathrm{Im}(X(t,k)) \end{pmatrix}. \quad (2)$$

This is possible, as real and imaginary parts are stored separately during the calculation of the DNN, which is also sketched in Figure 4, where pairs of nodes from layer 2 are connected with only one corresponding node in layer 3. Layer 3 represents the calculation of the magnitude and therefore halves the data size.

4) Logarithm: The last step is to form the logarithm of the squared magnitude $\chi = \log(|X(t,k)|^2)$, which is the common feature representation in speech recognition systems. It is easy to find its derivative as

$$\frac{\partial \chi}{\partial |X(t,k)|^2} = \frac{1}{|X(t,k)|^2}.$$

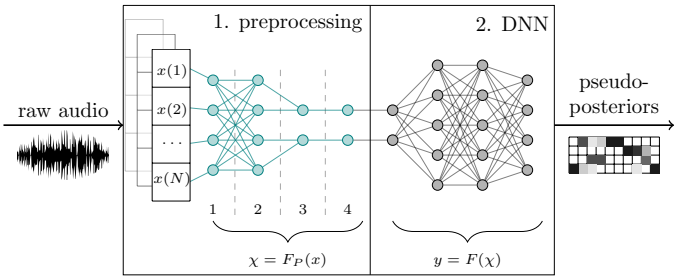


Fig. 4: For the creation of adversarial samples, we use an ASR system where the preprocessing is integrated into the DNN. Layers 1–4 represent the separate preprocessing steps. Note that this is only a sketch of the used DNN and that the used DNN contains far more neurons.

F. Hearing Thresholds

Psychoacoustic hearing thresholds allow us to limit audible distortions from all signal manipulations. More specifically, we use the hearing thresholds during the manipulation of the input signal in order to limit audible distortions. For this purpose, we use the original audio signal to calculate the hearing thresholds $\mathbf{H}$ as described in Section II-D. We limit the differences $\mathbf{D}$ between the original signal spectrum $\mathbf{S}$ and the modified signal spectrum $\mathbf{M}$ to the threshold of human perception for all times t and frequencies k

$$D(t,f) \leq H(t,k), \quad \forall t,k,$$
$$\text{with} \quad D(t,k) = 20 \cdot \log_{10} \frac{|S(t,k) - M(t,k)|}{\max_{t,k}(|\mathbf{S}|)}.$$

The maximum value of the power spectrum $|\mathbf{S}|$ defines the reference value for each utterance, which is necessary to calculate the difference in dB. Examples for $|\mathbf{S}|$, $|\mathbf{M}|$, $|\mathbf{D}|$, and $\mathbf{H}$ in dB are plotted in Figure 5, where the power spectra are plotted for one utterance.

We calculate the amount of distortion that is still acceptable via

$$\mathbf{\Phi} = \mathbf{H} - \mathbf{D}. \quad (3)$$

The resulting matrix $\mathbf{\Phi}$ contains the difference in dB to the calculated hearing thresholds.

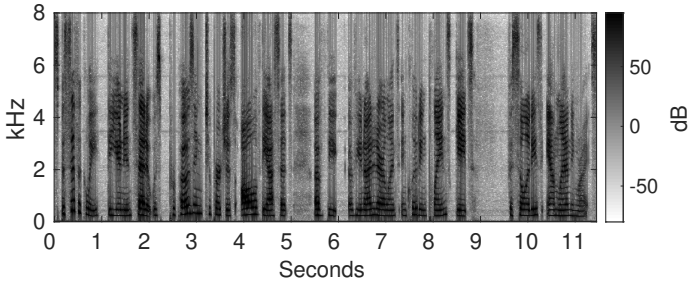
In the following step, we use the matrix $\mathbf{\Phi}$ to derive scaling factors. First, because the thresholds are tight, an additional variable λ is added, to allow the algorithm to differ from the hearing thresholds by small amounts

$$\mathbf{\Phi}^* = \mathbf{\Phi} + \lambda. \quad (4)$$

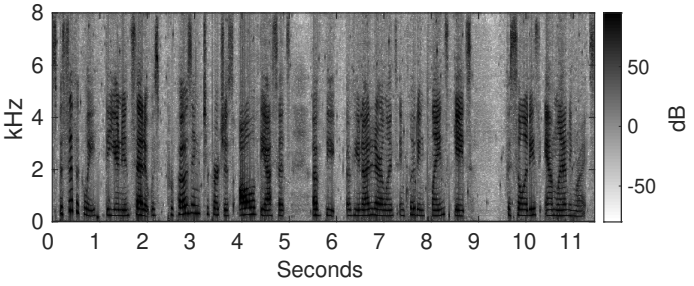
In general, a negative value for $\Phi^*(t,k)$ indicates that we crossed the threshold. As we want to avoid more noise for these time-frequency-bins, we set all $\Phi^*(t,k) < 0$ to zero. We then obtain a time-frequency matrix of scale factors $\hat{\mathbf{\Phi}}$ by normalizing $\mathbf{\Phi}^*$ to values between zero and one, via

$$\hat{\Phi}(t,k) = \frac{\Phi^*(t,k) - \min_{t,k}(\mathbf{\Phi}^*)}{\max_{t,k}(\mathbf{\Phi}^*) - \min_{t,k}(\mathbf{\Phi}^*)}, \quad \forall t,k.$$

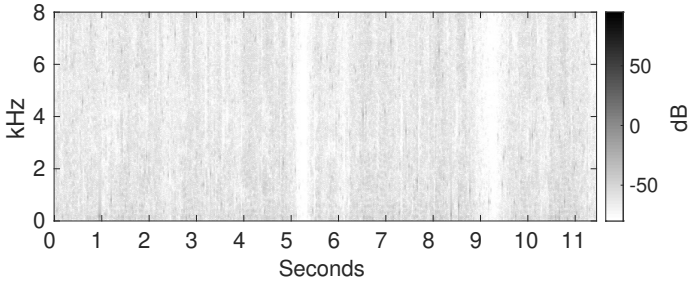
The scaling factors are applied during each backpropagation iteration. Using the resulting scaling factors $\hat{\Phi}(t,k)$ typically leads to good results, but especially in the cases where only



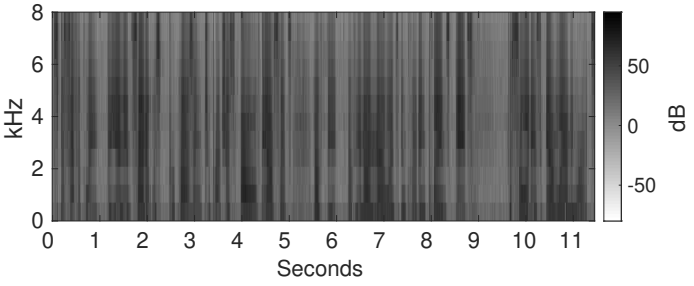
(a) Original audio signal power spectrum $|\mathbf{S}|$ with transcription: *"SPECIFICALLY THE UNION SAID IT WAS PROPOSING TO PURCHASE ALL OF THE ASSETS OF THE OF UNITED AIRLINES INCLUDING PLANES GATES FACILITIES AND LANDING RIGHTS"*



(b) Adversarial audio signal power spectrum $|\mathbf{M}|$ with transcription: *"DEACTIVATE SECURITY CAMERA AND UNLOCK FRONT DOOR"*



(c) The power spectrum of the difference between original and adversarial $|\mathbf{D}|$.



(d) Hearing thresholds $\mathbf{H}$.

Fig. 5: Original audio sample (5a) in comparison to the adversarial audio sample (5b). The difference of both signals is shown in Figure 5c. Figure 5d visualizes the hearing thresholds of the original sample, which are used for the attack algorithm.

very small changes are acceptable, this scaling factor alone is not enough to satisfy the hearing thresholds. Therefore, we use another, fixed scaling factor, which only depends on the hearing thresholds $\mathbf{H}$. For this purpose, $\mathbf{H}$ is also scaled to values between zero and one, denoted by $\hat{\mathbf{H}}$.

Therefore, the gradient $\nabla X(t,k)$ calculated via Equation (2) between the DFT and the magnitude step is scaled by both scaling factors

$$\nabla X^*(t,k) = \nabla X(t,k) \cdot \hat{\Phi}(t,k) \cdot \hat{H}(t,k), \quad \forall t,k.$$

IV. Experiments and Results

With the help of the following experiments, we verify and assess the proposed attack. We target the ASR system *Kaldi* and use it for our speech recognition experiments. We also compare the influence of the suggested improvements to the algorithm and assess the influence of significant parameter settings on the success of the adversarial attack.

A. Experimental Setup

To verify the feasibility of targeted adversarial attacks on state-of-the-art ASR systems, we have used the default settings for the *Wall Street Journal* (WSJ) training recipe of the *Kaldi* toolkit [38]. Only the preprocessing step was adapted for the integration into the DNN. The WSJ data set is well suited for large vocabulary ASR: it is phone-based and contains more than 80 hours of training data, composed of read sentences of the Wall Street Journal recorded under mostly clean conditions. Due to the large dictionary with more than $100,000$ words, this setup is suitable to show the feasibility of targeted adversarial attacks for arbitrary transcriptions.

For the evaluation, we embedded the hidden voice commands (i. e., target transcription) in two types of audio data: speech and music. We collect and compare results with and without the application of hearing thresholds, and with and without the use of forced alignment. All computations were performed on a 6-core Intel Core i7-4960X processor.

B. Metrics

In the following, we describe the metrics that we used to measure recognition accuracy and to assess to which degree the perturbations of the adversarial attacks needed to exceed hearing thresholds in each of our algorithm's variants.

1) Word Error Rate: As the adversarial examples are primarily designed to fool an ASR system, a natural metric for our success is the accuracy with which the target transcription was actually recognized. For this purpose, we use the Levenshtein distance [31] to calculate the word error rate (WER). A dynamic-programming algorithm is employed to count the number of deleted D, inserted I, and substituted S words in comparison to the total number of words N in the sentence, which together allows for determining the word error rate via

$$WER = \frac{D+I+S}{N}. \tag{5}$$

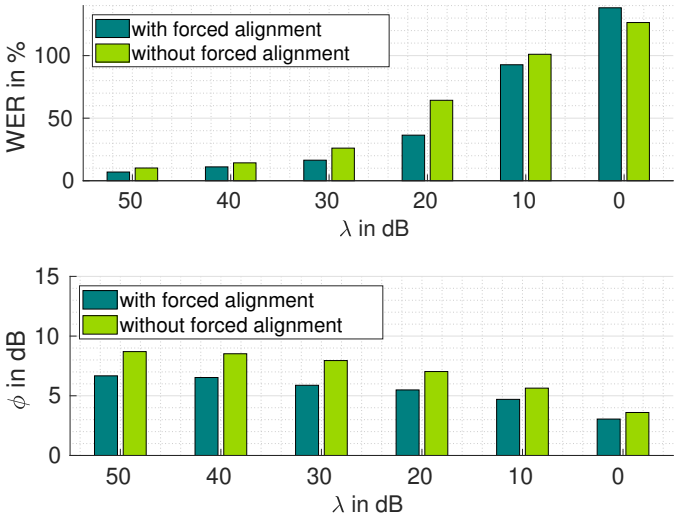


Fig. 6: Comparison of the algorithm with and without forced alignment, evaluated for different values of λ.

When the adversarial example is based on audio samples with speech, it is possible that the original text is transcribed instead of—or in addition to—the target transcription. Therefore, it can happen that many words are inserted, possibly even more words than contained in the target text. This can lead to WERs larger than 100 %, which can also be observed in Table I, and which is not uncommon when testing ASR systems under highly unfavorable conditions.

2) Difference Measure: To determine the amount of perceptible noise, measures like the signal-to-noise-ratio (SNR) are not sufficient given that they do not represent the subjective, perceptible noise. Hence, we have used $\mathbf{\Phi}$ of Equation (3) to obtain a comparable measure of audible noise. For this purpose, we only consider values > 0, as only these are in excess of the hearing thresholds. This may happen when λ is set to values larger than zero, or where changes in one frequency bin also affect adjacent bins.

We sum all values $\Phi(t,k) > 0$ for $t = 0,\dots,T-1$ and $k = 0,\dots,N-1$ and divide the sum by $T \cdot N$ for normalization. This value is denoted by ϕ. It constitutes our measure of the degree of perceptibility of noise.

C. Improving the Attack

As a baseline, we used a simplified version of the algorithm, forgoing both the hearing thresholds and the forced alignment stage. In the second scenario, we included the proposed hearing thresholds. This minimizes the amount of added noise but also decreases the chance of a valid adversarial example. In the final scenario, we added the forced alignment step, which results in the full version of the suggested algorithm, with a clearly improved WER.

For the experiments, a subset of 70 utterances for 10 different speakers from one of the WSJ test sets was used.

1) Backpropagation: First, the adversarial attack algorithm was applied without the hearing thresholds or the forced alignment. Hence, for the alignment, the audio sample was divided equally into the states of the target transcription. We used 500 iterations of backpropagation. This gives robust results and requires a reasonable time for computation. We chose a learning rate of 0.05, as it gave the best results during preliminary experiments. This learning rate was also used for all following experiments.

For the baseline test, we achieved a WER of 1.43 %, but with perceptible noise. This can be seen in the average ϕ, which was 11.62 dB for this scenario. This value indicates that the difference is clearly perceptible. However, the small WER shows that targeted attacks on ASR systems are possible and that our approach of backpropagation into the time domain can very reliably produce valid adversarial audio samples.

2) Hearing Thresholds: Since the main goal of the algorithm is the reduction of the perceptible noise, we included the hearing thresholds as described in Section III-F. For this setting, we ran the same test as before.

In this case, the WER increases to 64.29 %, but it is still possible to create valid adversarial samples. On the positive side, the perceptible noise is clearly reduced. This is also indicated by the much smaller value of ϕ of only 7.04 dB.

We chose $\lambda = 20$ in this scenario, which has been shown to be a good trade-off. The choice of λ highly influences the WER, a more detailed analysis can be found in Table I.

3) Forced Alignment: To evaluate the complete system, we replaced the equal alignment by forced alignment. Again, the same test set and the same settings as in the previous scenarios were used. Figure 6 shows a comparison of the algorithm's performance with and without forced alignment for different values of λ, shown on the x-axis. The parameter λ is defined in Equation (4) and describes the amount the result can differ from the thresholds in dB. As the thresholds are tight, this parameter can influence the success rate but does not necessarily increase the amount of noise. In all relevant cases, the WER and ϕ show better results with forced alignment. The only exception is the one case of $\lambda = 0$, where the WER is very high in all scenarios.

In the specific case of $\lambda = 20$, set as in Section IV-C2, a WER of 36.43 % was achieved. This result shows the significant advantage of the forced alignment step. At the same time, the noise was again noticeably reduced, with $\phi = 5.49$ dB. This demonstrates that the best temporal alignment noticeably increases the success rate in the sense of the WER, while at the same time reducing the amount of noise—a rare win-win situation in the highly optimized domain of ASR. In Figure 5, an example of an original spectrum of an audio sample is compared with the corresponding adversarial audio sample. One can see the negligible differences between both signals. The added noise is plotted in Figure 5c. Figure 5d depicts the hearing thresholds of the same utterance, which were used in the attack algorithm.

D. Evaluation

In the next steps, the optimal settings are evaluated, considering the success rate, the amount of noise, and the time required to generate valid adversarial examples.

TABLE I: WER in % for different values for λ in the range of $0\,\mathrm{dB}$ to $50\,\mathrm{dB}$, comparing speech and music as audio inputs.

	Iter.	None	50 dB	40 dB	30 dB	20 dB	10 dB	0 dB
Speech	500	2.14	6.96	11.07	16.43	36.43	92.69	138.21
	1000	1.79	3.93	5.00	7.50	22.32	76.96	128.93
Music	500	1.04	8.16	13.89	22.74	31.77	60.07	77.08
	1000	1.22	10.07	9.55	15.10	31.60	56.42	77.60

TABLE II: The perceptibility ϕ over all samples in the test sets in dB.

	Iter.	None	50 dB	40 dB	30 dB	20 dB	10 dB	0 dB
Speech	500	10.11	6.67	6.53	5.88	5.49	4.70	3.05
	1000	10.80	7.42	7.54	6.85	6.46	5.72	3.61
Music	500	4.92	3.92	3.56	3.53	3.39	2.98	2.02
	1000	5.03	3.91	3.68	3.40	3.49	3.20	2.30

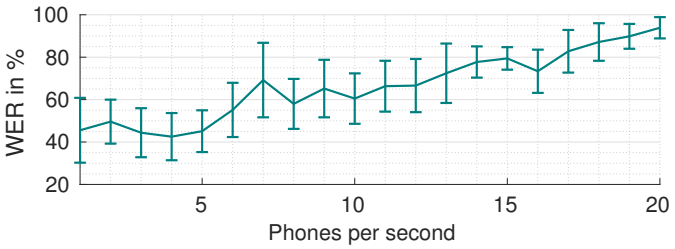


Fig. 7: Accuracy for different phone rates. To create the examples, 500 iterations of backpropagation and $\lambda = 20$ are used. The vertical lines represent the variances.

1) Evaluation of Hearing Thresholds: In Table I, the results for speech and music samples are shown for 500 and for 1000 iterations of backpropagation, respectively. The value in the first row shows the setting of λ. For comparison, the case without the use of hearing thresholds is shown in the column 'None.' We applied all combinations of settings on a test set of speech containing 72 samples and a test set of music containing 70 samples. The test set of speech was the same as for the previous evaluations and the target text was the same for all audio samples.

The results in Table I show the dependence on the number of iterations and on λ. The higher the number of iterations and the higher λ, the lower the WER becomes. The experiments with music show some exceptions to this rule, as a higher number of iterations slightly increases the WER in some cases. However, this is only true where no thresholds were employed or for $\lambda = 50$.

As is to be expected, the best WER results were achieved when the hearing thresholds were not applied. However, the results with applied thresholds show that it is indeed feasible to find a valid adversarial example very reliably even when minimizing human perceptibility. Even for the last column, where the WER increases to more than $100\,\%$, it was still possible to create valid adversarial examples, as we will show in the following evaluations.

In Table II, the corresponding values for the mean perceptibility ϕ are shown. In contrast to the WER, the value ϕ decreases with λ, which shows the general success of the thresholds, as smaller values indicate a smaller perceptibility. Especially when no thresholds are used, ϕ is significantly higher than in all other cases. The evaluation of music samples shows smaller values of ϕ in all cases, which indicates that it is much easier to conceal adversarial examples in music. This was also confirmed by the listening tests (cf. Section V).

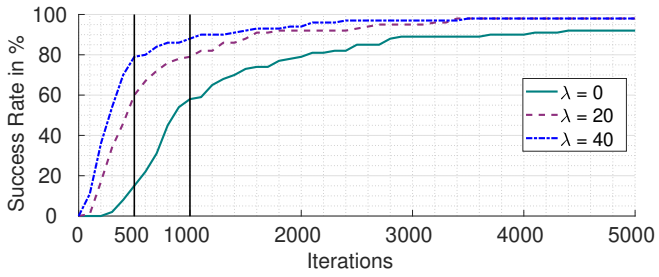
2) Phone Rate Evaluation: For the attack, timing changes are not relevant as long as the target text is recognized correctly. Therefore, we have tested different combinations of audio input and target text, measuring the number of phones that we could hide per second of audio, to find an optimum phone rate for our ASR system. For this purpose, different target utterances were used to create adversarial examples from audio samples of different lengths. The results are plotted in Figure 7. For the evaluations, 500 iterations and $\lambda = 20$ were used. Each point of the graph was computed based on 200 adversarial examples with changing targets and different audio samples, all of them speech.

Figure 7 shows that the WER increases clearly with an increasing phone rate. We observe a minimum for 4 phones per second, which does not change significantly at a smaller rate. As the time to calculate an adversarial sample increases with the length of the audio sample, 4 phones per second is a reasonable choice.

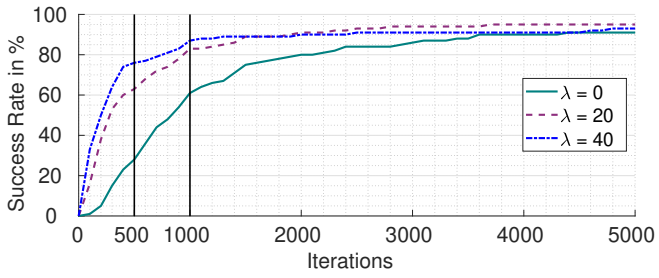
3) Number of Required Repetitions: We also analyzed the number of iterations needed to obtain a successful adversarial example for a randomly chosen audio input and target text. The results are shown in Figure 8. We tested our approach for speech and music, setting $\lambda = 0$, $\lambda = 20$, and $\lambda = 40$, respectively. For the experiments, we randomly chose speech files from 150 samples and music files from 72 samples. For each sample, a target text was chosen randomly from 120 predefined texts. The only constraint was that we used only audio-text-pairs with a phone rate of $6\,\mathrm{phones}$ per second or less, based on the previous phone rate evaluation. In the case of a higher phone rate, we chose a new audio file. We repeated the experiment 100 times for speech and for music and used these sets for each value of λ. For each round, we ran 100 iterations and checked the transcription. If the target transcription was not recognized successfully, we started the next 100 iterations and re-checked, repeating until either the maximum number of 5000 iterations was reached or the target transcription was successfully recognized. An adversarial example was only counted as a success if it had a WER of $0\,\%$. There were also cases were no success was achieved after 5000 iterations. This varied from only 2 cases for speech audio samples with $\lambda = 40$ up to 9 cases for music audio samples with $\lambda = 0$.

In general, we can not recommend using very small values of λ with too many iterations, as some noise is added during each iteration step and the algorithm becomes slower. Even though the results in Figure 8 show that it is indeed possible to successfully create adversarial samples with λ set to zero, but 500 or 1000 iterations may be required. Instead, to achieve a higher success rate, it is more promising to switch to a higher value of λ, which often leads to fewer distortions overall than using $\lambda = 0$ for more iterations. This will also be confirmed by the results of the user study, which are presented in Section V.

The algorithm is easy to parallelize and for a ten-second audio file, it takes less than two minutes to calculate the



(a) Speech



(b) Music

Fig. 8: Success rate as a function of the number of iterations. The upper plot shows the result for speech audio samples and the bottom plot the results for music audio samples. Both sets were tested for different settings of λ.

adversarial perturbations with 500 backpropagation steps on a 6-core (12 threads) Intel Core i7-4960X processor.

E. Comparison

We compare the amount of noise with *CommanderSong* [59], as their approach is also able to create targeted attacks using *Kaldi* and therefore the same DNN-HMM-based ASR system. Additionally, is the only recent approach, which reported she signal-to-noise-ratio (SNR) of their results.

The SNR measures the amount of noise σ, added to the original signal x, computed via

$$\text{SNR(dB)} = 10 \cdot \log_{10} \frac{P_x}{P_\sigma},$$

where P_x and P_σ are the energies of the original signal and the noise. This means, the *higher* the SNR, the *less* noise was added.

Table III shows the SNR for successful adversarial samples, where no hearing thresholds are used (None) and for different values of λ (40 dB, 20 dB, and 0 dB) in comparison to *CommanderSong*. Note, that the SNR does not measure the perceptible noise and therefore, the resulting values are not always consistent with the previously reported ϕ. Nevertheless, the results show, that in all cases, even if no hearing thresholds are used, we achieve higher SNRs, meaning, less noise was added to create a successful adversarial example.

V. User Study

We have evaluated the human perception of our audio manipulations through a two-part user study. In the transcription test, we verified that it is impossible to understand the voice command hidden in an audio sample. The MUSHRA test provides an estimate of the perceived audio quality of adversarial examples, where we tested different parameter setups of the hiding process.

TABLE III: Comparison of SNR with *CommanderSong* [59], best result shown in bold print.

	None	40 dB	20 dB	0 dB	*CommanderSong* [59]
SNR	15.88	17.93	**21.76**	19.38	15.32

A. Transcription Test

While the original text of a speech audio sample should still be understandable by human listeners, we aim for a result where the hidden command cannot be transcribed or even identified as speech. Therefore, we performed the *transcription test*, in which test listeners were asked to transcribe the utterances of original and adversarial audio samples.

1) Study Setup: Each test listener was asked to transcribe 21 audio samples. The utterances were the same for everyone, but with randomly chosen conditions: 9 original utterances, 3 adversarial examples with $\lambda = 0$, $\lambda = 20$, and $\lambda = 40$ respectively and 3 difference signals of the original and the adversarial example, one for each value of λ. For the adversarial utterances, we made sure that all samples were valid, such that the target text was successfully hidden within the original utterance. We only included adversarial examples which required ≤ 500 iterations.

We conducted the tests in a soundproofed chamber and asked the participants to listen to the samples via headphones. The task was to type all words of the audio sample into a blank text field without any provision of auto-completion, grammar, or spell checking. Participants were allowed to repeat each audio sample as often as needed and enter whatever they understood. In a post-processing phase, we performed manual corrections on minor errors in the documented answers to address typos, misspelled proper nouns, and numbers. After revising the answers in the post-processing step, we calculated the WER using the same algorithms as introduced in Section IV-B1.

2) Results: For the evaluation, we have collected data from 22 listeners during an internal study at our university. None of the listeners were native speakers, but all had sufficient English skills to understand and transcribe English utterances. As we wanted to compare the WER of the original utterances with the adversarial ones, the average WER of 12.52 % overall test listeners was sufficient. This number seems high, but the texts of the WSJ are quite challenging. For the evaluation, we ignored all cases where only the difference of the original and adversarial sample was played. For all of these cases, none of the test listeners was able to recognize any kind of speech and therefore no text was transcribed.

For the original utterances and the adversarial utterances, an average WER of 12.59 % and 12.61 % was calculated. The marginal difference shows that the difference in the audio does not influence the intelligibility of the utterances. Additionally, we have tested the distributions of the original utterances and

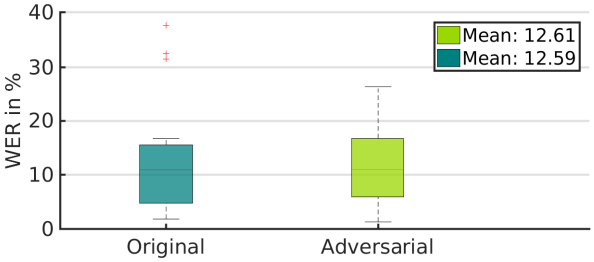


Fig. 9: WER for all 21 utterances over all test listeners of the original utterances and the adversarial utterances.

the adversarial utterances with a two-sided t-test to verify whether both distributions have the same mean and variance. The test with a significance level of 1 % shows no difference for the distributions of original and adversarial utterances.

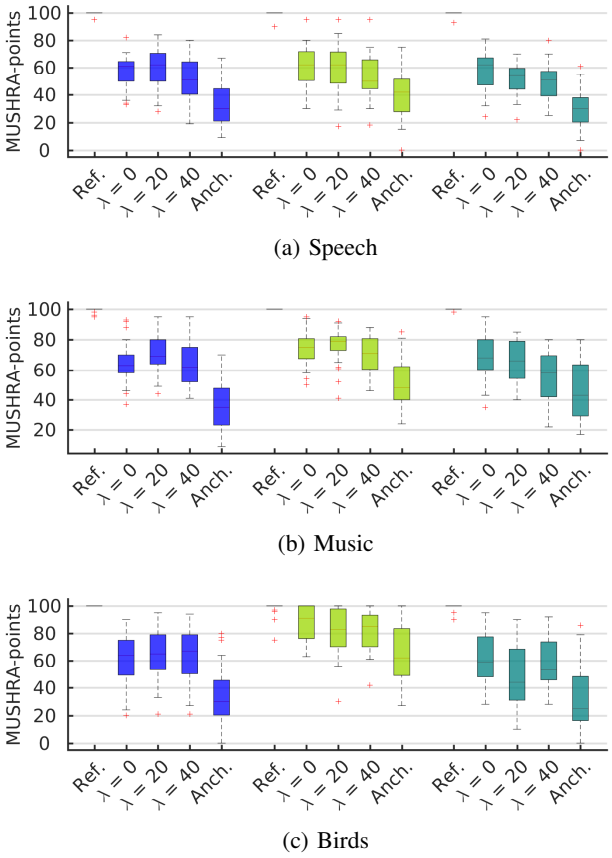
In the second step, we have also compared the text from the test listeners with the text which was hidden in the adversarial examples. For this, we have measured a WER far above 100 %, which shows that the hidden text is not intelligible. Also, there are only correct words which were in the original text, too, and in all cases these were frequent, short words like *is*, *in*, or *the*.

B. MUSHRA Test

In the second part of the study, we have conducted a Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) test, which is commonly used to rate the quality of audio signals [44].

1) Study Setup: The participants were asked to rate the quality of a set of audio signals with respect to the original signal. The set contains different versions of the original audio signal under varying conditions. As the acronym shows, the set includes a *hidden reference* and an *anchor*. The former is the sample with the best and the latter the one with the worst quality. In our case, we have used the original audio signal as the *hidden reference* and the adversarial example, which was derived without considering the hearing thresholds, as *anchor*. Both the *hidden reference* and the *anchor* are used to exclude participants, who were not able to identify either the *hidden reference* or the *anchor*. As a general rule, the results of participants who rate the *hidden reference* with less than 90 MUSRHA-points more than 15 % of the time are not considered. Similarly, all results of listeners who rate the *anchor* with more than 90 MUSRHA-points more than 15 % of the time are excluded. We used the *webMUSHRA* implementation, which is available online and was developed by AudioLabs [45].

We have prepared a MUSHRA test with nine different audio samples, three for speech, three for music, and three for recorded twittering birds. For all these cases, we have created adversarial examples for $\lambda = 0$, $\lambda = 20$, $\lambda = 40$, and without hearing thresholds. Within one set, the target text remained the same for all conditions, and in all cases, all adversarial examples were successful with ≤ 500 iterations. The participants were asked to rate all audio signals in the set on a scale between 0 and 100 (0–20: Bad, 21–40: Poor, 41–60: Fair, 61–80: Good, 81–100: Excellent). Again, the listening test was conducted in a soundproofed chamber and via headphones.



(a) Speech

(b) Music

(c) Birds

Fig. 10: Ratings of all test listeners in the MUSHRA test. We tested three audio samples for speech, music, and twittering birds. The left box plot of all nine cases shows the rating of the original signal and therefore shows very high values. The anchor is an adversarial example of the audio signal that had been created without considering hearing thresholds.

2) Results: We have collected data from 30 test listeners, 3 of whom were discarded due to the MUSHRA exclusion criteria. The results of the remaining test listeners are shown in Figure 10 for all nine MUSHRA tests. In almost all cases, the reference is rated with 100 MUSHRA-points. Also, the anchors are rated with the lowest values in all cases.

We tested the distributions of the anchor and the other adversarial utterances in one-sided t-tests. For this, we used all values for one condition overall nine MUSHRA tests. The tests with a significance level of 1 % show that in all cases, the anchor distribution without the use of hearing thresholds has a significantly lower average rating than the adversarial examples where the hearing thresholds are used. Hence, there is a clear perceptible difference between adversarial examples with hearing thresholds and adversarial examples without hearing thresholds.

During the test, the original signal was normally rated higher than the adversarial examples. However, it has to be considered that the test listeners directly compared the original signal with the adversarial ones. In an attack scenario, this would not be the case, as the original audio signal is normally

unknown to the listeners. Despite the direct comparison, there is one MUSRHA test where the adversarial examples with hearing thresholds are very frequently rated with a similar value as the reference and more than 80 MUSHRA-points. This is the case for the second test with twittering birds, which shows that there is a barely perceptible difference between the adversarial samples and the original audio signal.

Additionally, we observed that there is no clear preference for a specific value of λ. The samples with $\lambda = 0$ received a slightly higher average rating in comparison to $\lambda = 20$ and $\lambda = 40$, but there is only a significant difference for the distributions of $\lambda = 0$ and $\lambda = 40$. This can be explained with the different number of iterations, since, as shown in Section IV-D3, for a higher value of λ, fewer iterations are necessary and each iteration can add noise to the audio signal.

VI. Related Work

Adversarial machine learning techniques have seen a rapid development in the past years, in which they were shown to be highly successful for image classifiers. Adversarial examples have also been used to attack ASR systems, however, there, the modifications of the signals are usually quite perceptible (and sometimes even understandable) for human listeners. In the following, we review existing literature in this area and discuss the novel contributions of our approach.

A. Adversarial Machine Learning Attacks

There are many examples of successful adversarial attacks on image files in the recent past and hence we only discuss selected papers. In most cases, the attacks were aimed at classification only, either on computer images or real-world attacks. For example, Evtimov et al. showed one of the first real-world adversarial attacks [15]. They created and printed stickers, which can be used to obfuscate traffic signs. For humans, the stickers are visible. However, they seem very inconspicuous and could possibly fool autonomous cars. Athalye and Sutskever presented another real-world adversarial perturbation on a 3D-printed turtle, which is recognized as a rifle from almost every point of view [2]. The algorithm to create this 3D object not only minimizes the distortion for one image, but for all possible projections of a 3D object into a 2D image. A similar attack on a universal adversarial perturbation was presented by Brown et al. [7]. They have created a patch which works universally and can be printed with any color printer. The resulting image will be recognized as a toaster without covering the real content even partially. An approach which works for tasks other than classification is presented by Cisse et al. [11]. The authors used a probabilistic method to change the input signal and also showed results for different tasks but were not successful in implementing a robust targeted attack for an ASR system. Carlini et al. introduced an approach with a minimum of distortions where the resulting images only differ in a few pixels from the original files [9]. Additionally, they are robust against common distillation defense techniques [36].

Compared to attacks against audio signals, attacks against image files are easier, as they do not have temporal dependencies. Note that the underlying techniques for our attack are similar, but we had to refine them for the audio domain.

B. Adversarial Voice Commands

Adversarial attacks on ASR systems focus either on hiding a target transcription [8] or on obfuscating the original transcription [11]. Almost all previous works on attacks against ASR systems were not DNN-based and therefore use other techniques [8], [52], [61]. Furthermore, none of the existing attacks used psychoacoustics to hide a target transcription in another audio signal.

Carlini et al. have shown that targeted attacks against HMM-only ASR systems are possible [8]. They use an inverse feature extraction to create adversarial audio samples. However, the resulting audio samples are not intelligible by humans in most of the cases and may be considered as noise, but may make thoughtful listeners suspicious. A different approach was shown by Vaidya et al. [52], where the authors changed an input signal to fit the target transcription by considering the features instead of the output of the DNN. Nevertheless, the results show high distortions of the audio signal and can easily be detected by a human.

An approach to overcome this limitation was proposed by Zhang et al. They have shown that an adversary can hide a transcription by utilizing non-linearities of microphones to modulate the baseband audio signals with ultrasound above 20 kHz [61]. The main downside of the attack is the fact that the information of the necessary features needs to be retrieved from audio signal, recorded with the specific microphone, which is costly in practice. Furthermore, the modulation is tailored to a specific microphone an adversary wants to attack. As a result, the result may differ if another microphone is used. Song and Mittael [48] and Roy et al. [42] introduced similar ultrasound-based attacks that are not adversarial examples, but rather interact with the ASR system in a frequency range inaudible to humans.

Recently, Carlini and Wagner published a technical report in which they introduce a general targeted attack on ASR systems using CTC-loss [10]. The attack is based on a gradient-descent-based minimization [9] (as used in previous image classification adversarial attacks), but the loss function is represented via CTC-loss, which is optimized for time sequences. Compared to our approach, the perceptible noise level is higher and the attack is less effective, given that the algorithm needs a lot of time to calculate an adversarial example since it is based on a grid search.

CommanderSong [59] is also evaluated against Kaldi and uses backpropagation to find an adversarial example. However, in order to minimize the noise, approaches from the image domain are borrowed. Therefore, the algorithm does not consider human perception. Additionally, the attack is only shown for music and the very limited over-the-air attack highly depends on the speakers and recording devices as the attack parameters have to be adjusted especially for these components.

Our approach is different from all previous studies on adversarial perturbations for ASR, as we combine a targeted attack with the requirement that the added noise should be barely, if at all, perceptible. We use a modified backpropagation scheme, which has been very successful in creating adversarial perturbations for image classification and we initialize our optimization by forced-alignment to further minimize audible noise. Also, the psychoacoustics-based approach focuses

on the human-perceptible frequency range, which is different from ultrasound-based approaches.

C. Hiding Information in Audio

Watermarking approaches use human perception to hide information about an image, video, or audio clip within itself [4], [13], [55]. The purpose in the case of watermarking, however, differs from our method and steganography, as it is used for copyright protection. Watermarking uses algorithms to hide information in audio signals within the lower frequencies or also with help of a psychoacoustic model [46], [56]. Differently from watermarking and steganography, the frequency ranges cannot be chosen arbitrarily when an ASR system is to be attacked. Instead, the information must be presented in just those frequency regions, on which the ASR has been trained.

Audio steganography is motivated by the challenge of hiding additional information in an acoustic carrier signal, e. g., for transmitting sensitive information in case of comprehensive Internet censorship [23]. LSB techniques [1], [24] manipulate the binary representation of a signal and hide information in the least significant bits of a signal, which limits the perceived distortions to a minimum. In contrast to our work, such schemes ignore the acoustic characteristics of the carrier signal and achieve their hiding capabilities at the expense of disrupting the statistical characteristics of the original input. Modulation-based systems [23], [33] manipulate the carrier signal in the time or frequency domain to encode information within the signal characteristics. Such modulations allow the attacker to consider the frequency or energy features of a signal and help to provide a less conspicuous manipulation of the carrier signal. Both classes of steganography systems aim at hiding information in a legitimate carrier signal but are focused on creating a protected transmission channel within an untrusted system. In contrast, while our work is designed to provide a comparable level of *inconspicuousness*, we require the successful transcription of a target message through an automated speech recognition system, which precludes the use of watermarking or steganography algorithms here.

VII. Discussion

We have shown that it is possible to successfully attack state-of-the-art DNN-HMM ASR systems with targeted adversarial perturbations, which are barely or even impossible to distinguish from original audio samples. Based on different experiments, we demonstrated that it is possible to find the best setup for the proposed algorithm for the creation of adversarial examples. However, these results also open questions regarding possible countermeasures and future work.

A. Parameter Choice

The choice of the parameters highly affects the amount of perceptible noise. The evaluation has shown that a higher number of iterations increases the success rate, but simultaneously the amount of noise. However, for iterations < 500 the success rate is already very high and therefore, 500 should not be exceeded. Additionally, by this choice, the required calculation time is reduced as well. If the success rate needs to be raised, the increase of λ had a higher effect. Although the participants in the MUSHRA test did prefer smaller values for λ, there was no significant difference if λ was only increased by $20\,\mathrm{dB}$. Additionally, the phone rate should be set to an optimum value as this highly affects the success of the attack.

Besides improving the success of the attack, the choice of the original audio sample greatly influences the quality of the adversarial example. There might be use cases, where the original audio sample is fixed, but in general, the choice of the original sample is free. We recommend using music or other unsuspicious audio samples, like bird twittering, which do not contain speech, as speech has to be obfuscated, typically leading to larger required adversarial perturbations.

The process can be parallelized and is relatively fast in comparison to other attacks proposed in the past, as we have integrated the preprocessing into the backpropagation. Therefore, we recommend to use different promising setups and to choose that one which sounds the most inconspicuous while giving the required success rate.

B. Countermeasures

Distillation was shown to be a successful countermeasure against attacks in image classification [36]. It is a technique to improve the robustness of classification-based DNNs [19], which uses the output of a pre-trained DNN as soft labels in order to train a second DNN with these soft labels as targets. However, the transcription of the DNN-based ASR system not only depends on the classification result but also on the temporal alignment. Therefore, distillation might not be an appropriate countermeasure for ASR systems.

A general countermeasure could be to consider human perception. A very simple version would be to apply MP3 encoding to the input data. However, the DNN is not trained for that kind of data. Nevertheless, we did run some tests on our adversarial examples. With this setup, the original transcription could not be recovered, but the target transcription was also distorted. We assume that training the ASR-DNN with MP3-encoded audio files will only move the vulnerability into the perceptible region of the audio files, but will not circumvent blind spots of DNNs completely.

A more appropriate solution might be the adaption of the loss function during the training of the DNN. It can be utilized to measure a human-like perception-based difference. This may result in an ASR system, which is more similar to human speech recognition and, therefore, it is likely that hiding messages is more difficult.

C. Future Work

One obvious question is whether our attack also works in a real-world setup. For real-world attacks, different circumstances have to be considered. This mainly includes the acoustic transfer function, which is relevant if an audio signal is transferred over the air. Also, additional noise from different sources can be present in a real-world environment. In a controlled environment (e. g., an elevator) it is possible to calculate the transfer function and to consider or exclude external noise. Additionally, it is not unusual to have music in an elevator, which would make such a setup very unsuspicious. A transfer function independent approach may be borrowed from the image domain, specifically from Athalye et al. [2]. In this work, the loss function is designed in order to consider all

kinds of rotations and translations. Similar to their approach, the loss function of the ASR-DNN may not only be designed to optimize one specific transfer function but to maximize the expectation over all kinds of transfer functions. However, for a realistic over-the-air attack, we think it is worth it to spend more time investigating all possible attack vectors and their limitations. In this work, we, instead, wanted to focus on the general and theoretical feasibility of psychoacoustic hiding and therefore leave over-the-air attack for future work.

A similar attack can also be imagined for real applications, e. g., Amazon's Alexa. However, the detailed architecture of this system is hard to access and it requires a-priori investigations to obtain that kind of information. It may be possible to retrieve the model parameters with different model stealing approaches [22], [34], [37], [49], [54]. For Alexa, our reverse engineering results of the firmware indicate that Amazon uses parts of *Kaldi*. Therefore, a limited knowledge about the topology and parameters might be enough to create a model for a black-box attack. A starting point could be the keyword recognition of commercial ASR systems, e. g., "*Alexa*." This would have the advantage that the keyword recognition runs locally on the device and would, therefore, be easier to access.

For image classification, universal adversarial perturbations have already been successfully created [7], [29]. For ASR systems, it is still an open question if these kinds of adversarial perturbations exist and how they can be created.

VIII. Conclusion

We have presented a new method for creating adversarial examples for ASR systems, which explicitly take dynamic human hearing thresholds into account. In this way, borrowing the mechanisms of MP3 encoding, the audibility of the added noise is clearly reduced. We perform our attack against the state-of-the-art *Kaldi* ASR system and feed the adversarial input directly into the recognizer in order to show the general feasibility of psychoacoustics-based attacks.

By applying forced alignment and backpropagation to the DNN-HMM system, we were able to create inconspicuous adversarial perturbations very reliably. In general, it is possible to hide any target transcription within any audio file and, with the correct attack vectors, it was possible to hide the noise below the hearing threshold and make the changes psychophysically almost imperceptible. The choice of the original audio sample, an optimal phone rate, and forced alignment give the optimal starting point for the creation of adversarial examples. Additionally, we have evaluated different algorithm setups, including the number of iterations and the allowed deviation from the hearing thresholds. The comparison with another approach by Yuan et al. [59], which is also able to create targeted adversarial examples, shows that our approach needs far lower distortions. Listening tests have proven that the target transcription was incomprehensible for human listeners. Furthermore, for some audio files, it was almost impossible for participants to distinguish between the original and adversarial sample, even with headphones and in a direct comparison.

Future work should investigate the hardening of ASR systems by considering psychoacoustic models, in order to prevent these presently fairly easy attacks. Additionally, similar attacks should be evaluated on commercial ASR systems and in real-world settings (e.g., in a black-box setting or via over-the-air attacks).

Acknowledgments

This work was supported by Intel as part of the Intel Collaborative Research Institute "Collaborative Autonomous & Resilient Systems" (ICRI-CARS). In addition, this work was supported by the German Research Foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the States - EXC 2092 CaSa.

References

[1] M. Asad, J. Gilani, and A. Khalid, "An enhanced Least Significant Bit modification technique for audio steganography," in *Conference on Computer Networks and Information Technology*. IEEE, Jul. 2011, pp. 143–147.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *CoRR*, vol. abs/1707.07397, pp. 1–18, Jul. 2017.

[3] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," 2018.

[4] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE transactions on image processing*, vol. 10, no. 5, pp. 783–791, 2001.

[5] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, Nov. 2010.

[6] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Symposium on Information, Computer and Communications Security*. ACM, Mar. 2006, pp. 16–25.

[7] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *CoRR*, vol. abs/1712.09665, pp. 1–6, Dec. 2017.

[8] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. A. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Security Symposium*. USENIX, Aug. 2016, pp. 513–530.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Symposium on Security and Privacy*. IEEE, May 2017, pp. 39–57.

[10] ——, "Audio adversarial examples: Targeted attacks on Speech-to-Text," *CoRR*, vol. abs/1801.01944, pp. 1–7, Jan. 2018.

[11] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," *CoRR*, vol. abs/1707.05373, pp. 1–12, Jul. 2017.

[12] M. Cissé, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *Conference on Machine Learning*. PMLR, Aug. 2017, pp. 854–863.

[13] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[14] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Workshop on Security and Privacy in Smartphones & Mobile Devices*. ACM, Nov. 2014, pp. 63–74.

[15] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, pp. 1–11, Jul. 2017.

[16] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning*, vol. 107, no. 3, pp. 481–508, Mar. 2018.

[17] M. Fujimoto, "Factored deep convolutional neural networks for noise robust speech recognition," *Proc. Interspeech*, pp. 3837–3841, 2017.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, pp. 1–11, Dec. 2014.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, pp. 1–9, Mar. 2015.

[20] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *CoRR*, vol. abs/1804.08598, pp. 1–10, Apr. 2018.

[21] ISO, "Information Technology – Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbits/s – Part3: Audio," International Organization for Standardization, ISO 11172-3, 1993.

[22] M. Juuti, S. Szyller, A. Dmitrenko, S. Marchal, and N. Asokan, "PRADA: Protecting against DNN model stealing attacks," *CoRR*, vol. abs/1805.02628, pp. 1–16, May 2018.

[23] K. Kohls, T. Holz, D. Kolossa, and C. Pöpper, "SkypeLine: Robust hidden data transmission for VoIP," in *Asia Conference on Computer and Communications Security*. ACM, May 2016, pp. 877–888.

[24] J. Liu, K. Zhou, and H. Tian, "Least-Significant-Digit steganography in low bitrate speech," in *International Conference on Communications*. IEEE, Jun. 2012, pp. 1133–1137.

[25] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *CoRR*, vol. abs/1611.02770, 2016.

[26] D. Lowd and C. Meek, "Adversarial learning," in *Conference on Knowledge Discovery in Data Mining*. ACM, Aug. 2005, pp. 641–647.

[27] V. Manohar, D. Povey, and S. Khudanpur, "JHU kaldi system for arabic MGB-3 ASR challenge using diarization, audiotranscript alignment and transfer learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop*, Dec 2017, pp. 346–352.

[28] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: trainable text-speech alignment using kaldi," in *Proceedings of interspeech*, 2017, pp. 498–502.

[29] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Conference on Computer Vision and Pattern Recognition*. IEEE, Jul. 2017, pp. 86–94.

[30] T. Moynihan, "How to keep Amazon Echo and Google Home from responding to your TV," Feb. 2017, https://www.wired.com/2017/02/keep-amazon-echo-google-home-responding-tv/, as of December 17, 2018.

[31] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, Mar. 2001.

[32] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2015, pp. 427–436.

[33] M. Nutzinger, C. Fabian, and M. Marschalek, "Secure hybrid spread spectrum system for steganography in auditive media," in *Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, Oct. 2010, pp. 78–81.

[34] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical Black-Box attacks against machine learning," in *Asia Conference on Computer and Communications Security*. ACM, Apr. 2017, pp. 506–519.

[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *European Symposium on Security and Privacy*. IEEE, Mar. 2016, pp. 372–387.

[36] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Symposium on Security and Privacy*. IEEE, May 2016, pp. 582–597.

[37] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, "Transferability in machine learning: From phenomena to Black-Box attacks using adversarial samples," *CoRR*, vol. abs/1605.07277, pp. 1–13, May 2016.

[38] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Workshop on Automatic Speech Recognition and Understanding*. IEEE, Dec. 2011.

[39] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993.

[40] S. Ranjan and J. H. Hansen, "Improved gender independent speaker recognition using convolutional neural network based bottleneck features," *Proc. Interspeech 2017*, pp. 1009–1013, 2017.

[41] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "A network of deep neural networks for distant speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4880–4884.

[42] N. Roy, H. Hassanieh, and R. Roy Choudhury, "BackDoor: Making microphones hear inaudible sounds," in *Conference on Mobile Systems, Applications, and Services*. ACM, Jun. 2017, pp. 2–14.

[43] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," *CoRR*, vol. abs/1703.02136, pp. 1–7, Mar. 2017.

[44] N. Schinkel-Bielefeld, N. Lotze, and F. Nagel, "Audio quality evaluation by experienced and inexperienced listeners," in *International Congress on Acoustics*. ASA, Jun. 2013, pp. 6–16.

[45] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webMUSHRA – A comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, Feb. 2018.

[46] J. W. Seok and J. W. Hong, "Audio watermarking for copyright protection of digital audio data," *Electronics Letters*, vol. 37, no. 1, pp. 60–61, 2001.

[47] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: Increasing local stability of supervised models through robust optimization," *Neurocomputing*, 2018.

[48] L. Song and P. Mittal, "Inaudible voice commands," *CoRR*, vol. abs/1708.07238, pp. 1–3, Aug. 2017.

[49] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *USENIX Security Symposium*. USENIX, Aug. 2016, pp. 601–618.

[50] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi openkws system: Improving low resource keyword search," *Proc. Interspeech*, pp. 3597–3601, Aug. 2017.

[51] P. Upadhyaya, O. Farooq, M. R. Abidi, and Y. V. Varshney, "Continuous hindi speech recognition model based on kaldi asr toolkit," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, March 2017, pp. 786–789.

[52] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine Noodles: Exploiting the gap between human and machine speech recognition," in *Workshop on Offensive Technologies*. USENIX, Aug. 2015.

[53] J. Villalba, N. Brümmer, and N. Dehak, "Tied variational autoencoder backends for i-vector speaker recognition," *Interspeech, Stockholm*, pp. 1005–1008, Aug. 2017.

[54] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Symposium on Security and Privacy*. IEEE, May 2018.

[55] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, 1999.

[56] S. Wu, J. Huang, D. Huang, and Y. Q. Shi, "Self-synchronized audio watermark in dwt domain," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 5, May 2004, pp. V–V.

[57] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, Dec. 2017.

[58] ——, "The microsoft 2016 conversational speech recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5255–5259.

[59] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Security Symposium*. USENIX, Aug. 2018, pp. 49–64.

[60] V. Zantedeschi, M.-I. Nicolae, and A. Rawat, "Efficient defenses against adversarial attacks," in *Workshop on Artificial Intelligence and Security*. ACM, Nov. 2017, pp. 39–49.

[61] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Conference on Computer and Communications Security*. ACM, Oct. 2017, pp. 103–117.

[62] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, 3rd ed. Springer, 2007.