applied sciences

MDPI

Article

Evading Logits-Based Detections to Audio Adversarial Examples by Logits-Traction Attack

Songshen Han *, Kaiyong Xu, Songhui Guo, Miao Yu and Bo Yang

Third Academic, Information Engineering University, Zhengzhou 450001, China
* Correspondence: thordlos@foxmail.com

Abstract: Automatic Speech Recognition (ASR) provides a new way of human-computer interaction. However, it is vulnerable to adversarial examples, which are obtained by deliberately adding perturbations to the original audios. Thorough studies on the universal feature of adversarial examples are essential to prevent potential attacks. Previous research has shown classic adversarial examples have different logits distribution compared to normal speech. This paper proposes a logit-traction attack to eliminate this difference at the statistical level. Experiments on the LibriSpeech dataset show that the proposed attack reduces the accuracy of the LOGITS NOISE detection to 52.1%. To further verify the effectiveness of this approach in attacking detection based on logits, three different features quantifying the dispersion of logits are constructed in this paper. Furthermore, a richer target sentence is adopted for experiments. The results indicate that these features can detect baseline adversarial examples with an accuracy of about 90% but cannot effectively detect Logits-Traction adversarial examples, proving that Logits-Traction attack can evade the logits-based detection method.

Keywords: automatic speech recognition; speech processing; logits; audio adversarial attack; adversarial example detection

check for updates

Citation: Han, S.; Xu, K.; Guo, S.; Yu, M.; Yang, B. Evading Logits-Based Detections to Audio Adversarial Examples by Logits-Traction Attack. *Appl. Sci.* **2022**, *12*, 9388. https://doi.org/10.3390/app12189388

Academic Editor: Javier Hernando

Received: 20 July 2022
Accepted: 14 September 2022
Published: 19 September 2022

1. Introduction

The deep neural network has brought many conveniences to our daily life. However, due to adversarial examples, deploying deep neural networks in practical manufacturing or decision-making tasks is risky. Research on adversarial examples began in Computer Vision (CV), and Szegedy C et al. [1] found that adding subtle perturbations to a normal image could cause an incorrect model output. The study on speech adversarial examples borrows ideas from the CV domain but focuses more on attacking sequential models such as the Recurrent Neural Network (RNN) [2]. As Automatic Speech Recognition (ASR) models are deployed on systems that directly affect safety, such as smart homes and autonomous driving, research on speech adversarial examples is increasingly crucial.

Speech recognition models typically separate speech input into frames, map them to letter or word categories frame by frame, and then assemble them into sentences by decoding algorithms [3]. Adversarial examples against ASR usually perform targeted attacks by adding subtle perturbations to normal speech so that it is transformed into the text specified by the attacker, thus covertly executing malicious commands [4] and posing a greater security threat.

Researchers have proposed several methods for detecting speech adversarial examples. Samizade et al. [5] developed a CNN-based speech cepstral feature classification network. Still, the method has poor generalization performance in detecting adversarial examples generated by different methods and unknown methods. Rajaratnam et al. [6] investigated adding a large amount of random noise to a specific frequency band of the audio signal. Kwon et al. [7] modified normal audio by adding noise, low-pass/high-pass filtering, etc. Both studies exploited the property that adversarial examples have lower robustness than normal audio. However, the effectiveness of these methods in defending against complex attacks and over long speech requires further investigation. Yang et al. [8] segmented the speech data and calculated the consistency of the segment recognition results with the whole sequence recognition results to detect adversarial examples, but the location of the segmentation points was not universal.

The above detection methods focus on the differences between normal data and adversarial examples, but they do not incorporate the inference results of the model. There is more information in the judgment by the ASR model. For example, Park N et al. [9] exploited the difference in the logit distribution of normal speech and adversarial examples. The method can achieve an accuracy of over 95% and detect adversarial examples generated by multiple attack methods without task flow changes or additional training.

However, it is found that this detection method is susceptible to adaptive attacks. The key insight is that Park N et al. [9], who exploited only one feature to detect adversarial examples. The feature uses the difference between the maximum value and the next largest value in each frame of logits to quantify the dispersion degree. To attack this feature, this paper designs a new loss function term to construct Logits-Traction attacks. The specific process is shown in Figure 1.

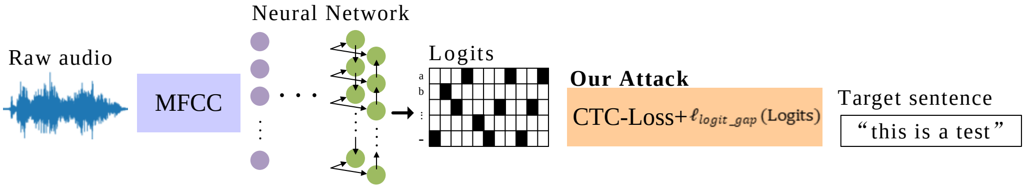


Figure 1. Logits-Traction attack.

The generation of speech adversarial examples is a multi-step iterative [10] optimization procedure. In each iteration, the original loss Connectionist Temporal Classification-Loss (CTC-Loss) adds adversarial perturbation that enables the adversarial example to be transcribed in the direction of the target sentence. Meanwhile, the new loss function term $\updownarrow_{logit\_gap}(Logits)$ increases the intra-frame logits difference, and they are not in a completely antagonistic relationship. Thus, it is possible to make the new adversarial example closer to the original speech in the evaluation metric of intra-frame difference, thereby evading the detection mechanism of Park N et al.

However, the difference of logits between normal speech and adversarial examples has a spatial distribution. The detection method [9] only measures this difference in one dimension. The similarity of this quantitative metric does not imply that our adversarial examples have the same logits distribution as normal speech. If alternative evaluation metrics [11] can detect our novel adversarial examples, the adversarial example detection method based on the dispersion of logits is still valid when the attacker is unaware of the specific quantitative measure. To further investigate whether an attack method generates adversarial examples with the same logit distribution as normal speech, this paper proposes logits-based detection methods that focus on three different features, including the intra-frame mean difference, decision frame variance, and the number of delineation statistics of logits.

The main contributions of this paper are summarized as follows:

This paper analyzed three high-precision speech adversarial examples detection methods. Among them, the Logits-based detection method is the most promising because it does not need prior knowledge when detecting adversarial examples. To evade such detection, this paper also studies two representative logits in different ASR models, and further analyzes the manifestations and causes of logits distribution differences.

To quantify this difference, this paper defined logits dispersion to describe the quality of logits. A smaller dispersion of logits implies that one speech is more likely to be an adversarial example. This paper proposes a Logits-Traction attack to improve the logits dispersion. It constructs a new loss function to generate adversarial examples, and this attack is compatible with the logit types of different ASR models.

This paper extends the logit features exploited by the detection method to evaluate the effectiveness of the above attack method. It focuses on three different features, which can detect Carlini and Wagner (C&W) adversarial samples with high accuracy, proving the effectiveness of these features. These features are then used to detect the Logits-Traction adversarial examples. The experimental results show that the detection method based on these features has different degrees of accuracy decline when detecting the Logits-Traction adversarial examples, which leads to the failure of the detection method. Logit-Traction attacks can effectively avoid detection based on logits.

2. Related Work

2.1. Audio Adversarial Attack

The classical approach for performing target adversarial attacks against ASR was proposed by Carlini & Wagner et al. [12], who constructed a white-box attack against the DeepSpeech [13] model. By evaluating the model parameters and calculating the Connectionist Temporal Classification (CTC) loss [3] from logits to the target phrase, the adversarial perturbation on the entire waveform was optimized iteratively, and the targeted attack was realized on arbitrary audio waveforms. Yakura et al. [14] continued the above idea. To produce an adversarial example with impacts in the physical world, they incorporated the effects of real-world noise into the optimization process. Then, the decline of adversarial examples in the real world was simulated by band-pass filtering, impulse response, and white Gaussian noise. The adversarial examples with lower loss function scores after decay are expected to be more effective in the real world. The C&W attack generates smaller adversarial perturbations but can still result in a non-negligible noise floor. Qin Yao et al. [15] further optimized the auditory masking principle of psychoacoustics introduced by Schönherr [16]. They optimized the adversarial examples obtained from the C&W attack [12] method by focusing the adversarial perturbations on audio regions that humans would not notice.

The optimization process against perturbations is more difficult when the network parameters of the target model are not accessible. Taori et al. [17] used a two-stage attack algorithm: a genetic algorithm [18] in the first stage and gradient estimation [19] in the fine-tuning stage. However, when the original speech and target text are long, the attack algorithm is less efficient and may even fail.

When the discrepancy between the adversarial examples and the original speech needs to be modest, the attacker should perform several queries to the target model. The adversarial perturbation is reduced by decreasing the perturbation factor [20] after each successful attack. The C&W attack [12] is the baseline of white-box audio adversarial attacks.

2.2. Audio Adversarial Example Detection

In operating systems that receive commands from the transcription results of ASRs [21], such as smart home and autonomous driving systems, detection or defense modules are employed to safeguard the system against targeted adversarial attacks. The most researched methods for detecting audio adversarial examples include spectrogram detection, time-dependent detection, and logits difference-based detection.

2.2.1. Spectrogram-Based Detection

The noise introduced by the C&W attack that the human ear cannot ignore can be depicted more clearly in the frequency domain. Samizade et al. [5] used a convolutional neural network to classify the speech spectrum according to the differences in the Mel-frequency cepstral coefficients (MFCC) features [22] of the adversarial examples. When known attack methods are adopted to generate adversarial examples for training, the models trained with white-box C&W [12] and black-box Alzantot [23] adversarial examples achieve a detection accuracy of over 98% for their respective types of adversarial examples.

However, a model trained with only Alzantot adversarial examples [23] cannot detect C&W adversarial examples. When applied to adversarial examples generated in other ways, the spectrogram-based detection approach shows an inadequate generalization performance.

2.2.2. Time-Dependent Detection

The ASR model converts speech data to text with obvious temporal correspondence. In their study of normal speech data, Yang et al. [8] discovered good consistency between the results of transcription of speech fragments and those of whole sentences. However, in adversarial examples, there is a large difference. Taking advantage of this characteristic allows time-dependent detection to detect adversarial examples at a low cost. In addition, knowing the attack method and many adversarial examples is no longer necessary, demonstrating excellent generalizability. However, a rough setting of the intercept ratio will lead to an unacceptable false-positive rate due to the irregular silences in the speech data.

2.2.3. Logits Difference Detection

Originally from computer vision, Wang Y et al. [24] and Aigrain J et al. [25] discovered the divergence of logit distribution between the adversarial examples and normal images. They [24] trained a Long Short-Term Memory (LSTM) network to analyze logit sequence features in the semantic space. This method does not require understanding the model details or modifying the network and can detect adversarial examples with a success rate of more than 90%. However, each speech is divided into tens of frames for the ASR task [26], while the logit structure of each frame is identical to that of the image classification task. Park N et al. [9] used the logit statistical feature difference between the original speech and the adversarial example, added noise to the logits of the speech to be detected and observed the changes in the transcription results. This method has a high detection rate for white-box [12], black-box [17] and physical attack [14] adversarial examples.

The above detection method usually judges all speech to be detected. However, this paper is more concerned with the voice that smart homes or autonomous driving systems will soon accept. They are either derived from normal speech or adversarial examples carefully constructed by the attacker to perform the targeted attack. The command execution system will automatically reject transcribed speeches as meaningless text. Our approach assumes that adversarial examples that perform target and nontarget attacks have different logit feature distributions. If this assumption is true, concentrating on the identification of adversarial examples intended to perform targeted attacks can increase the detection success rate.

3. Proposed Method

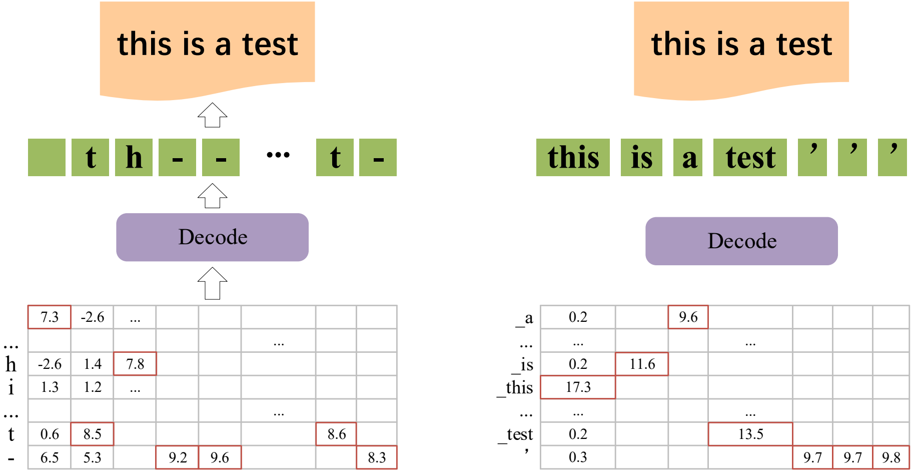
To improve the similarity between adversarial examples and original speech in models with different loss functions, this paper investigates two representative logits types in ASR systems, namely sparse logits and dense logits. Then, this paper proposes the concept of logit dispersion and designs a new loss function term for the characteristics of each type of logit so that the adversarial examples are optimized in the direction of increasing logit dispersion. To explore further whether the adversarial examples generated by the innovative method can approach the logits of normal speech in several quantitative dimensions, this paper proposes three different features to calculate logits dispersion. These features are used to verify whether an adaptive attack against one metric method can simultaneously reduce the detection accuracy of three other features and show the distribution characteristics of adversarial example logits under different attack methods.

3.1. Logits Type in the ASR Model

Logits are a vector that represents the raw outputs [27] of the neural network. In the model performing the image classification task, logits are one-dimensional sequences followed by a SoftMax layer [28]. In the ASR task, the output of RNN is logits [29]. It is a two-dimensional vector, which is usually transcribed into a character vector or word vector using beam search decoding and then assembled into a complete sentence.

In practice, the logits structure of the ASR model [13,30] can be represented as [batch_size, frames, classes], where "batch_size" represents the number of speeches in each iteration. "Frames" represents the number of frames of each speech, and is dependent on the length of the speech sequence and the MFCC window size. "Classes" represents the categories into which each frame may be transcribed.

In character-level transcription (Figure 2a), "A–Z", "Space", and "Blank" pseudo-characters (represented by "-") are included in the transcription classes. "Blank" means no characters and can only be used to separate two characters. For differentiation, this paper refers to the frames with the highest value in the "Blank" category as the "blank frame" (frame 4, 5, and the last frame on Figure 2a), and the remaining frames are called the "decision frame". In the process of generating adversarial examples, the conversion between "blank frame" and "decision frame" is more accessible than that between "decision frame" and "decision frame". Meanwhile, it tends to get a more significant intra-frame difference. This paper defines the logits structure of alternating "blank frame" and "decision frame" as sparse logits. ASR models using CTC loss functions, such as DeepSpeech [13], typically includes sparse logits. During the optimization of adversarial examples, the inter-frame interactions of the sparse logit are minimal, and the faults generally are exhibited as mis-transcribed characters in words.



(**a**) (**b**)

Figure 2. The schematic diagram of sparse logits (**a**) and dense logits (**b**).

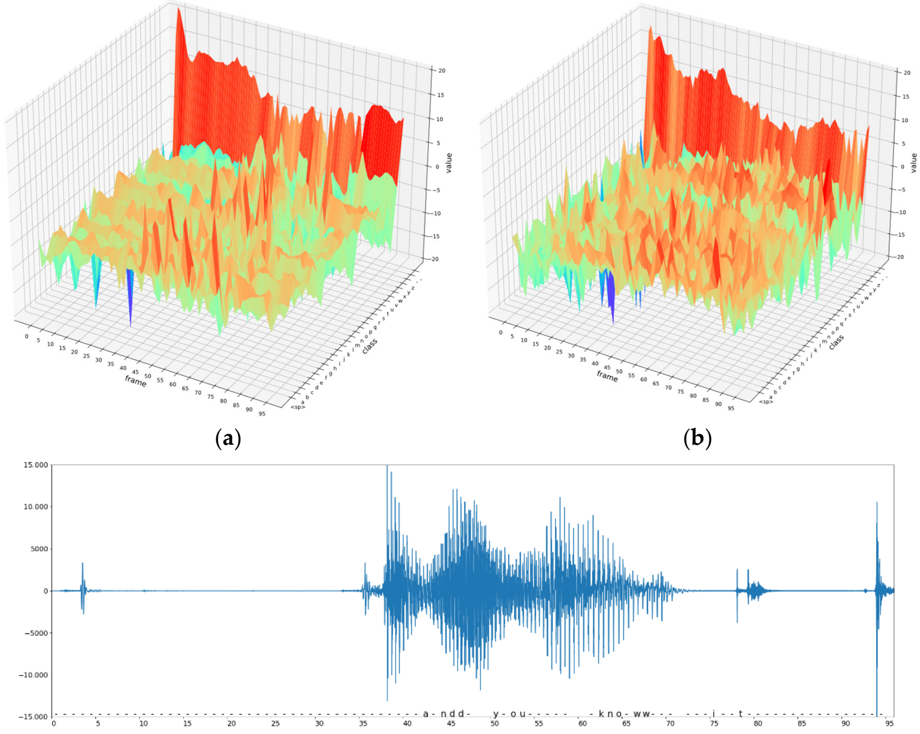
In word-level transcription (Figure 2b), the transcribed classes are related to specific models, including all labelled words and word roots from the training data. The transcription of succeeding frames requires the results of the previous frames, and the output of the last state in the RNN network contains the logits information of the whole speech sequence. In this logits type, the first d "decision frames" ($d = 4$ on the Figure 2b) determine the transcription result of the speech with a substantial intra-frame difference. In contrast, the subsequent $frames - d$ frames are meaningless "blank frames" with a near-zero intra-frame difference. This paper defines such logits with distinct bounded frames as dense logits. Dense logits are included in ASR models such as Lingvo [30] that employ cross-entropy loss functions. Its incorrect transcription results can only originate from classes, and the mistakes are presented as successive words or root transcription errors.

Therefore, the distribution of blank frames in dense and sparse logits is significantly different. The blank frame cannot effectively affect the transcription result compared to the character frame. More weight should be given to the character frame to improve the detection accuracy. Indeed, the logit distribution of original speech and adversarial example are more different in these frames.

3.2. Logits Value of Adversarial Examples

To visually show the difference in logits between the original speech and adversarial examples, this paper plots a three-dimensional representation of the logits of the original speech and adversarial examples derived from it. As shown below, the frame axis in the figure represents the MFCC frame of the speech vector. The number of coordinate points on this axis is proportional to the length of the speech audio. The class axis represents probability distribution, where the model decodes each frame into all characters, and that probability is unnormalized. The value axis represents the specific value of the above possibility. The class that has the maximum value in each frame is transcribed characters. So the transcription task can also be considered a frame-by-frame classification task. Below (a) are the logits of the original speech whose transcription result is “and you know”, and (b) are the logits of the C&W adversarial example generated based on the above original speech and whose transcription result is “This is a test”. The logits were output by the DeepSpeech model. This paper connects these points as surfaces to better show their changing process.

Figure 3a depicts the logits value of the original speech sequence, and Figure 3c depicts the raw signal of the original speech. In it, frames 0 to 35 represent silence. The “blank” characters (denoted by “-”, the last column in 3.a class axis) score significantly higher than other classes, indicating that DeepSpeech is highly confident that these frames are transcripted to “blank”. Frames 35 to 80 are the stages of human vocalization. In these stages, each frame column has a better value on only one category, with logit values that are much lower and less changeable for other classes. In the original speech, the confidence of “blank frame” and “decision frame” is generally higher. The Figure 3b represents the adversarial example, except for the “blank” class in the “blank frame”, which has a more significant logit value. All other logit values are minor. In the “decision frame”, each frame column has multiple large values, and DeepSpeech only recognizes them as the target transcription with weak confidence.



(**a**) (**b**)

(**c**)

Figure 3. Logits of the original speech (**a**), adversarial example (**b**) and raw speech signal (**c**).

To quantify this difference numerically, Park N et al. proposed a detection method based on the “distribution of the difference between the maximum value of logit and the second largest value” by adding Gaussian noise to the logits. Their method assumes that frames with a minor difference are more susceptible to maximum bit reversal when the noise follows the same distribution, leading to altered transcription results. As shown in Figure 3, the speech with more altered transcriptional effects after adding noise is more likely to be an adversarial example. After calculating the number of erroneous characters caused by adding noise to the logit, the error threshold can be determined at a statistical level, and speech adversarial example detection independent of the network structure is implemented.

This paper defines the degree of disparity between the values of the classes in the “decision frame” of logits as the logit dispersion. In dense logits, it is sufficient to calculate by the values of “decision frames” that have the potential to affect the final transcription result. However, in sparse logits, the values of the “blank” pseudo-characters class in all “decision frames” must be removed.

3.3. Logits-Traction Adversarial Example Generation

3.3.1. Inspiration

In computer vision, the C&W L2 attack [31] for the image classification task sets the confidence in the one-dimensional logit sequence to calculate the intra-frame difference feature. The loss term to perform the targeted attack is:

$$f(x') = max\left(max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa\right) \tag{1}$$

where x' represents the adversarial example, $Z(x')_i$ represents the logit value of the $i-th$ class, t is the target class, and κ is the confidence of the adversarial example. To achieve target attacks, a larger κ is more likely to cause model misclassification.

Such a loss function is a piecewise function, and it works well in classification tasks. The same mechanism in speech adversarial examples will be affected by logits frame [32,33], and it is impractical to set a κ value for each frame. This paper only comprehensively considers the intra-frame logits difference of all frames, and generates adversarial examples with an overall higher logit intra-frame difference based on the fact that adversarial examples are always optimized to decrease the loss function value.

3.3.2. Threat Model

For a given speech signal sequence x and target transcribed phrase k, this paper constructs another audio waveform $x' = x + \delta$ so that $f(x') = k$, and the logits dispersion of the generated adversarial examples is close to normal speech. To generate adversarial examples, the network structure and parameters of the target model are completely evaluated, and then they are optimized and tested in the digital realm. This is a typical tactic for white-box attacks.

3.3.3. Logits-Traction Loss Function

In both image and audio, generating adversarial examples against a white-box model can be considered as updating δ via gradient descent [34,35] on a predefined loss function:

$$\mathcal{L}(x, \delta, t) = \updownarrow_{model}(f(x+\delta), t) + c\cdot\updownarrow_{metric}(x, x+\delta) \tag{2}$$

where $\updownarrow_{model}$ denotes the loss function of the target model and $\updownarrow_{metric}$ denotes the distortion measure function. Parameter c weighs the relative importance of achieving adversarial attacks and reducing adversarial perturbations.

Due to the interframe interaction and the additional decoding steps in ASR, it is not easy to construct a loss that corresponds to the optimization objective.

Since the improvement of the loss does not need to consider the interaction between frames and additional decoding steps, the literature designs a loss function term, that improves the attack efficiency and the robustness of adversarial examples. Our method draws on this and aims to construct adversarial examples with new features. In our method, the optimization objective is extended as follows:

$$\mathcal{L}(x,\delta,t) = \updownarrow_{model}(f(x+\delta),t) + c_1 \cdot \updownarrow_{logit\_gap}\Big(f_{logits}(x+\delta)\Big) + c_2 \cdot \updownarrow_{metric}(\delta) \tag{3}$$

In the process of backpropagation, this loss function generates $\Delta\delta$ that achieves target transcription while producing a minimal $\updownarrow_{logit\_gap}$. The formal description of $\updownarrow_{logit\_gap}$ will be elaborated below.

Our optimization objective is composed of the following three components:

1. $\updownarrow_{model}$ denotes the original loss function used in the ASR model. The loss of sparse logits is CTC-Loss, and the loss of dense logits is cross-entropy loss function. Using the same loss function as the original model to generate adversarial examples is based on the following observation:

$$\updownarrow_{model}(f(x+\delta),t) \leq 0 \Rightarrow f(x\prime) = t \tag{4}$$

2. $\updownarrow_{logit\_gap}$ measures the logits value gap of the original output, calculates the difference between the maximum value and the second-highest value in all categories for each frame, and then adds these values together. The specific calculation method is shown in Algorithm 1.

Algorithm 1: *logit_gap* Measurement Method

Input: $f_{logits}(x+\delta)\ \Big(logits\ \in \mathbb{R}^{(b,n,m)}\Big)$, the original output of the adversarial example $x+\delta$ passing through the ASR model during each iteration.
Output: *gap_sum*, a vector describing the intra-frame difference of logits.

```
initialize $gap\_sum$
for $i = 0 : b$
    initialize $gap$
    for $j = 0 : n$
        $fst\_class = max\Big(logits^{(i,j,:)}\Big)$
        set $argmax\Big(logits^{(i,j)}\Big) = 0$
        $scd\_class = max\Big(logits^{(i,j,:)}\Big)$
        $gap+ = scd\_class - fst\_class$
    $gap\_sum[i] = gap$
return $gap\_sum$
```

where b denotes batch_size, n denotes frames, and m denotes classes; "0:b" means traversing over values from 0 to b, and ":" means it traverses all the values of the dimension in symbol resides (*ie.* 0 to b). A smaller $\updownarrow_{logit\_gap}$ means that the neural network has higher confidence in the current transcription result.

3 $\updownarrow_{metric}$ measures adversarial perturbations, and it is defined as:

$$\updownarrow_{metric}(\delta) = \sum_{i=0}^{n} \delta_i^2 \tag{5}$$

This paper adds the term $\updownarrow_{logit\_gap}$ to filter the C&W adversarial examples further. This may make the adversarial perturbations satisfy both constraints, resulting in speech distortion. Therefore, $\updownarrow_{metric}$ is added to limit the disturbance by penalizing adversarial perturbations with large absolute values.

3.4. Detection Based on Logits Dispersion by Other Three Features

This section further explores the effectiveness of our attack method. Given that the logit distribution of adversarial examples differs from that of normal speech, this paper designs detection mechanisms that focus on different features. The statistical features of logits in several other dimensions are calculated, such as intra-frame mean difference, decision frame variance, and the number of delineation statistics. It is checked whether Logits-Traction adversarial examples affect the detection accuracy based on these features.

As described in Section 3.2, the number of "decision frames" in dense logits is small, and each frame has too many possible categories. Adding noise to the logits by Park N et al. cannot distinguish between normal speech and adversarial examples. Another drawback of this method is that the random noise mechanism may produce inconsistent findings for numerous detections of the same speech. To this end, this paper designs new detection methods, and the detection system deployment [36] is shown in Figure 4:

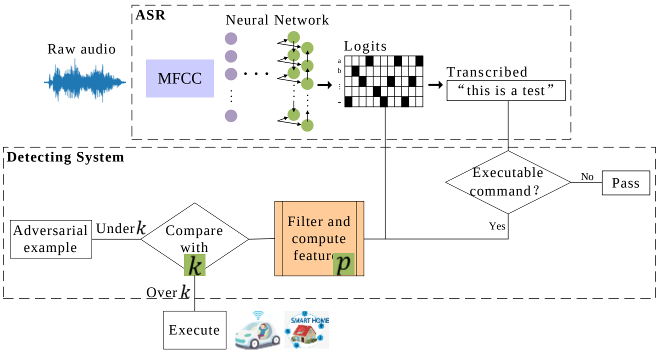
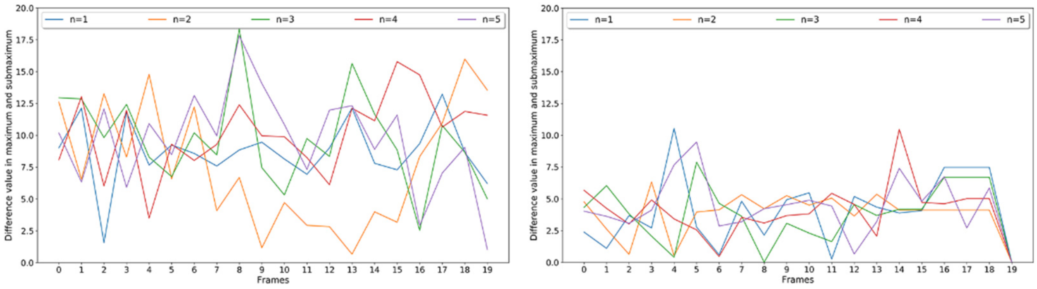


Figure 4. Deployment of the detection system.

This detection system consists of three fixed logits feature calculation algorithms, a hyperparameter p and a threshold k. p and k are only related to the protected model. When a text transcribed by ASR is recognized as an executable command, the detection system performs feature calculation on its logits. At first, the algorithm screens out logit values that satisfy the p condition, then applies the feature calculation algorithm to acquire the evaluation value and compares this value with the threshold k. Those exceeding the range of k are considered adversarial examples.

3.4.1. Detection Based on the Intra-Frame Mean Difference

Figure 3 illustrates that the difference between the maximum value and the next largest value in logits differs for normal speech and adversarial examples. The dense logits also show the identical pattern as shown in Figure 5:



(**a**) (**b**)

Figure 5. Difference of each frame in Lingvo by five normal speech (**a**) and C&W adversarial examples (**b**).

These five adversarial examples are generated from the code [37] provided in the literature [15]. It indicates that adversarial examples have a smaller intra-frame difference than normal speech. This paper quantifies this difference by calculating the sum between the maximum value and the next largest value for all decision frames. The phenomenon that long text speech with a significant difference sum is more likely to be judged as normal speech can be avoided by dividing by the number of decision frames:

$$\Delta_{12} = \frac{\sum_{i \in D_{(n-t)}} \left(l_i^{max} - l_i^{sec} \right)}{(n-t)} \tag{6}$$

where $D_{(n-t)}$ denotes the set of serial numbers of character frames and l_i^{max} denotes the maximum value of the $i - th$ frame. The obtained Δ_{12} is compared with the threshold k, and those less than k are judged as adversarial examples.

3.4.2. Detection Based on the Variance of Filtered Values

The optimization objective of adversarial examples is to introduce an adversarial perturbation to yield the target expression. However, it does not suppress its original expression. The original expression may shift and spread in the adversarial perturbation. That will result in adversarial examples with slight logit variance. Focusing on this issue, this paper designs a statistical detection method based on the variance of filtered values so that under the same calculation method, the logit variance of adversarial examples can be smaller, and the logit variance of normal speech can be larger. The filter mechanism can distinguish adversarial examples from normal speech more clearly.

In the sparse logit ASR model, values greater than the hyperparameter p in the speech logits to be detected must be screened, and the variance of the non-negative values in matrix $A_{(n-t)\times(m-1)}$ is derived as the logit evaluation index:

$$\sigma^{+2} = \frac{1}{(n-t)\times(m-1)-r} \sum_{i=1}^{(n-t)} \sum_{j=1}^{(m-1)} \left(a_{ij} - \overline{x}^{+} \right)^2 \tag{7}$$

where $a_{ij} = \begin{cases} x_{ij}, \ x \geq p \\ \overline{x}^{+}, \ x < p \end{cases}$, $\overline{x}^{+}$ denotes the average of all values greater than p in this non-sparse matrix $A_{(n-t)\times(m-1)}$

$$\overline{x}^{+} = \frac{1}{(n-t)\times(m-1)-r} \sum_{i=1}^{(n-t)} \sum_{j=1}^{(m-1)} x_{ij}^{+} \tag{8}$$

where $x_{ij}^{+} = \begin{cases} x_{ij}, \ x \geq p \\ 0, \ x < p \end{cases}$, r denotes the number of all values greater than p in matrix $A_{(n-t)\times(m-1)}$.

In the dense logit ASR model, the variance of non-negative values in matrix $A_{(n-t)\times m}$ is calculated as the logit evaluation index, and the calculation method is the same as the above formula. Finally, σ^{+2} is compared with the statistical threshold k, and those less than the threshold are judged as adversarial examples.

3.4.3. Detection Based on the Number of Delineation Statistics

This method is still based on the assumptions introduced in Section 3.4.2 but is not concerned with the fluctuation of values. The following Figure 6 shows the distribution interval statistics of logits values of a speech and its adversarial examples in the Lingvo model. The original transcription of the speech is "THE MORE SHE IS ENGAGED IN HER PROPER DUTIES THE LESS LEISURE WILL SHE HAVE FOR IT EVEN AS AN ACCOMPLISHMENT AND A RECREATION", and the target sentence to generate the antagonistic example is "OLD WILL IS A FINE FELLOW BUT POOR AND HELPLESS SINCE MISSUS ROGERS HAD HER ACCIDENT".

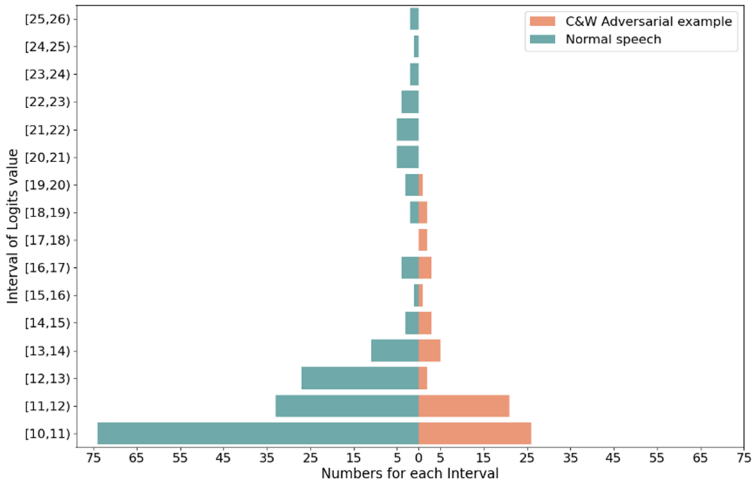


Figure 6. Logits value statistics in Lingvo.

It has a similar pattern to Figure 3. That is, the magnitude of logits value of adversarial examples is generally less than that of normal speech. This method just counts the number of values in the decision frames that are bigger than the filtered value p: $c^{+} = r$.

The speech is judged as an adversarial example if c^{+} is less than a threshold value. However, this method is easily affected by the number of decision frames and can only be used in adversarial example detection with similar text lengths.

4. Experimental Evaluation

To verify the effectiveness of our attack, the code [38] in the paper [9] is reproduced, the Logits-Traction (LT) adversarial examples are tested, and their detection accuracy is compared with that of C&W adversarial examples. Then, to evaluate the efficacy of our proposed detection features, C&W adversarial examples are built on multiple models and datasets. They are combined with the original speech data to create a new dataset and be tested. Finally, the accuracy of the above detection in detecting novel adversarial examples is tested to demonstrate that Logits-Traction adversarial examples can evade the logits-based detection features.

4.1. Dataset and Experiment Setup

Target ASR model: The target model of this paper for the experiments covers two types of logits. For sparse logits, the DeepSpeech model with the CTC loss function is selected, and for dense logits, the Lingvo model with the cross-entropy loss function is selected.

Source dataset: (1) LibriSpeech (Libri) [39]. The LibriSpeech dataset is from the LibriVox project. It is composed of English speech data with a sampling rate of 16 kHz, and the recording environment is relatively stable. This paper randomly selects speech data on the “test-clean” branch. (2) Mozilla Common Voices (MCV) [40]. The MCV dataset is a public speech dataset contributed by volunteers worldwide. The recording environment is inconsistent, and the volume levels vary. To be consistent with LibriSpeech, this paper uses the “sox” tool to downsample each speech from 48 kHz to 16 kHz, and randomly select speech data from it.

Dataset in our experiment: This paper randomly selects 600 speeches from each dataset as the original speech. The number of transcribed words ranges from 3 to 10, so the speech durations are approximately the same as those in the original papers [9]. However, only five target phrases are set in that study, leading to monotonous logits distribution of the adversarial example, thus making the success rate of our attack unrealistically high. To truly reflect the strength of our attack against the detection method, this paper randomly selects 100 transcriptions from 600 speeches as target phrases. Then, using the remaining

500 speeches as original speeches, the C&W adversarial example dataset and logits-Traction adversarial example dataset were created on DeepSpeech and Lingvo models, respectively. The success rate of detection based on three features is evaluated by C&W adversarial example dataset. The decline in detection success rate is evaluated on the LT adversarial example dataset.

This paper adopts accuracy, false-positive rate (FPR), and false-negative rate (FNR) as evaluation indicators. The accuracy rate is used to evaluate the quality of the detection method; FPR represents the proportion of the original speech detected as an adversarial example; FNR represents the proportion of adversarial examples detected as the original speech. FPR and FNR are negatively correlated due to the usage of threshold division for judgment.

4.2. Logits Traction Attack Effectiveness Evaluation

Since the original detection method can only detect sparse logits, his paper tests the Logits-Traction attack on the DeepSpeech v0.4.0 system. When constructing the adversarial example dataset, the Adam [41] optimizer is used. The learning rate is set to 100, and the maximum number of iterations is set to 1000. This paper performs detection on the two established adversarial datasets with the same configuration. The experimental results are as follows:

It can be seen from Table 1 that the FNR is 37.6% when detecting C&W adversarial examples, indicating that the detection method [9] has a limited detection success rate when the target phrases of the adversarial examples are abundant. In this more realistic configuration, LT attacks reduce the success rate of the LOGITS NOSIE detection method, proving that our LT adversarial examples are difficult to detect, thus evading the logits-based detection.

Table 1. Comparison of detection accuracy (%) on the LOGITS NOISE method.

Attack	**Performance**	**Libri**	**MCV**
C&W	Accuracy	81.6	66.1
	FPR	2.6	48.6
	FNR	37.6	39.2
LT **(Ours)**	Accuracy	52.1	28.5
	FPR	2.6	48.6
	FNR	93.7	94.4

Meanwhile, an excessive FPR for raw speech on the MCV dataset can be observed. After auditioning and analyzing the misclassified original speech, it is found that they are generally of poor quality or low volume and being chaotically transcribed by DeepSpeech. Since the detection system is deployed in the location shown in Figure 5, where logits are detected after transcription and before command execution, the high FPR of the original voice will not affect the regular operation of the detection system.

4.3. Logits Dispersion Detection Effectiveness Evaluation

When the detection algorithm is deployed, the defender only has information about the model to be protected, and it does not know the data source of the original speech used by the attacker. Therefore, the hyperparameter p and threshold k set by each detection algorithm should be fixed, and only change with the protected model. The current test parameters are set as follows:

1. Logit-Δ_{12} detection. It does not require a hyperparameter p. If the average difference between the largest and second largest values exceeds the threshold k, it is determined that the logits belong to the original speech;
2. Logit-σ^{+2} detection. In DeepSpeech, p is set to 0, which means to calculate the variance of the numbers whose logit value is greater than 0 in all decision frames. If

the variance is greater than the threshold k, it is determined that the logits belong to the original speech. Similarly, p is set to 10 in Lingvo;

3. Logit-c^+ detection. In DeepSpeech, p is set to 9, which means to count the number of values greater than 9 in all decision frames. If the number is greater than the threshold k, it is determined that the logits belong to the original speech. Similarly, p is set to 14 in Lingvo.

The detection results of the three detection methods on the C&W adversarial examples are presented in Table 2:

Table 2. Performance (%) of our three features.

Attack	Performance	DeepSpeech		Lingvo	
		Libri	MCV	Libri	MCV
Logit − Δ_{12}	Threshold	$k = 6$		$k = 6$	
	Accuracy	89.7	61.9	97.4	78.1
	FPR	8.4	47.4	5.2	41
	FNR	12.1	28.4	0	1.8
Logit − σ^{+2}	Threshold	$k = 12$		$k = 12$	
	Accuracy	91.1	58.5	96.7	75.8
	FPR	10.4	63.6	2.6	35.6
	FNR	7.5	18.3	4.0	12.3
Logit − c^+	Threshold	$k = 11$		$k = 8$	
	Accuracy	89.7	62.9	88.3	82.5
	FPR	6.0	46.4	7.4	23.6
	FNR	14.5	27.3	16.0	11.1

It can be seen from Table 2 that all three features have high accuracy on the LibriSpeech dataset, and the three features in the DeepSpeech and Lingvo models have similar detection accuracy distributions, indicating that our three detection methods based on logit dispersion are efficient. We take the detection using σ^{+2} features on the Lingvo system as an example, which achieved the highest detection accuracy on the LibriSpeech dataset. Their detection results are shown in Figure 7:

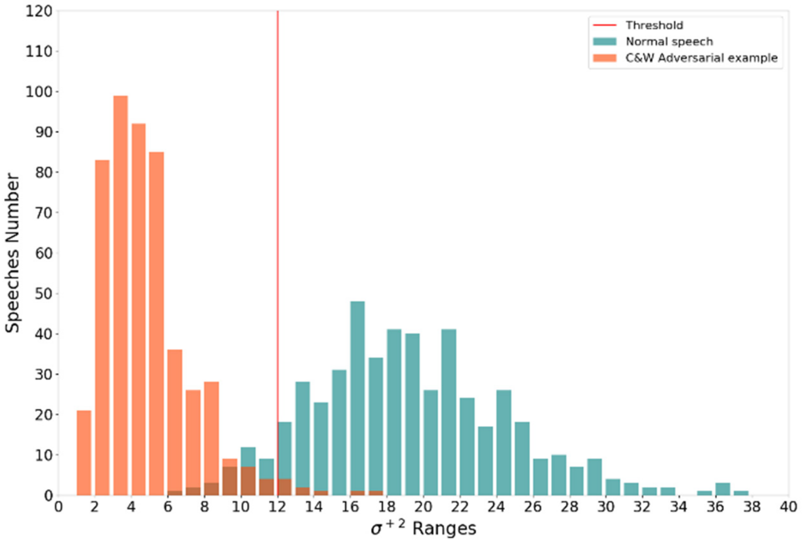


Figure 7. Detecting C&W adversarial examples by σ^{+2} feature in Lingvo.

The σ^{+2} of C&W adversarial examples less than 12 can be correctly detected. A few normal speech values less than 12 are mistakenly judged as adversarial examples. There is a clear dividing line between this two. The threshold $k = 12$, is the σ^{+2} dividing line between the adversarial example and normal speech.

Second, this paper uses the MCV dataset as comparison experiments, and its FNR is lower, meaning that our detection features can effectively detect even adversarial examples generated based on chaotic speech. Finally, the detection accuracy between our three features is similar. This is because the feature of difference, variance, and the absolute number of values are not entirely independent in the calculation. These features are all affected by the more significant value in logit decision frames.

Meanwhile, several experiments show that the detection accuracy cannot be improved effectively by tuning the hyperparameter p and the threshold k. That is an inherent drawback of these three features in quantifying logit dispersion. They cannot draw a clear demarcation line between the original data and the adversarial examples, and false and missed detections always exist.

4.4. Effectiveness of Logits Traction Attack against Our Three Detections

In this experiment, the values of the parameter p and threshold k obtained from each feature in the above experiments are preserved. These features are used to detect LT adversarial examples. This section only focuses on the FNR decline caused by LT adversarial examples, which means that the proportion of adversarial examples is not detected by the above features, Table 3:

Table 3. Three features detect LT adversarial examples experiment.

Attack	**Performance**	**DeepSpeech**		**Lingvo**	
		Libri	**MCV**	**Libri**	**MCV**
Logit $-\Delta_{12}$	Threshold	$k=6$		$k=6$	
	FNR	57.8	65.0	39.2	21.4
Logit $-\sigma^{+2}$	Threshold	$k=12$		$k=12$	
	FNR	56.1	59.6	41.4	31.5
Logit $-c^{+}$	Threshold	$k=11$		$k=8$	
	FNR	61.2	55.4	18.0	15.1

The experimental results show that the Logits-Traction adversarial examples can evade the logit dispersion-based detection. A false negative rate of more than 50% at the statistical level makes these three features no longer practical. The quantified logits dispersion of Logits-Traction adversarial examples has a more similar distribution to that of the original speech. As shown in Figure 8, there is no reasonable threshold to distinguish adversarial examples from the original speech.

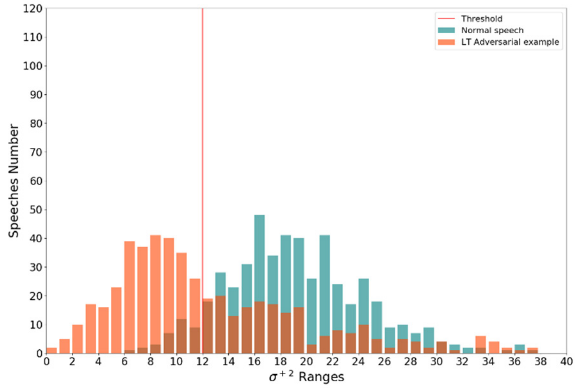


Figure 8. Detecting LT adversarial examples by σ^{+2} feature in Lingvo.

Meanwhile, compared with DeepSpeech, the effect of the Logit-Traction attack on the Logit-c^{+} feature in Lingvo is not obvious. This result indicates that the c^{+} feature of LT adversarial is still inferior to that of the original speech. Due to the expression in the original speech, the logits value of the adversarial example has difficulty in reaching the same level. That is the native advantage of the Lingvo system. Customizing the adversarial example detection system for Lingvo can affect the number of larger values in logits.

The experiments in Sections 4.3 and 4.4 mean that our Logit-traction attack will bring the logit distribution of the antagonistic example closer to the ideal Logit distribution of normal speech. However, the attack success rate is also affected by the characteristics of interest in the detection method.

5. Discussion

This paper performed an adaptive attack to the detection based on logit dispersion to prove its feasibility. The advantage of this method is that it can generate adversarial examples with similar logits distribution to normal speech. At least the current detection based on logits cannot detect the adversarial examples effectively. Moreover, these experiments give us some enlightenment. To protect the human-computer interaction system based on ASR, defenders should further investigate the methods using logits features for adversarial example detection. As mentioned in Section 4.4, the study of logit features with advantages in adversarial example detection for a specific model still has broad application prospects. Meanwhile, since the threshold is statistically obtained from the dataset, the detection method using threshold division cannot reasonably deal with the logits feature near that point. Designing a computational approach for discriminating adversarial examples near the threshold can further improve detection accuracy based on logit dispersion.

Considering the influence of computer vision, researchers usually use the perturbation size to evaluate speech adversarial examples. Adversarial examples with minor adversarial perturbations seem more normal and are less likely to be noticed. However, this is not enough. Our detection fully demonstrates that the adversarial examples generated based on classical C&W methods have universal logit features, and they are easily detected if no measures are taken. In addition, we also notice that the Logits-Traction adversarial example introduces more noise than the C&W adversarial example. This paper quantifies this drawback by comparing the Signal-to-noise ratio (SNR) of the Logits-Traction adversarial example with the C&W adversarial example. The calculation method is shown in Equation (9).

$$SNR = 10log_{10}\frac{P_x}{P_\delta} \tag{9}$$

where P_x and P_δ mean energy of raw audio and added adversarial perturbation, respectively. Using the Librispeech dataset, we calculated the average SNR of the above C&W adversarial examples and Logits-traction adversarial examples with the same speech to the same target phrase. The average SNR of C&W adversarial examples was -6.605. The average SNR of Logits-Traction adversarial examples was -7.299. This means that the Logits-Traction method introduces more adversarial perturbation. This is caused by the more restrictive adversarial example search space of the method in this paper. It seems that the logits dispersion is inversely proportional to the size of the adversarial perturbation, and we will explore this conjecture further in subsequent studies. We suggest that the logit dispersion should be used as one of the evaluation indicators in the follow-up research. The logits dispersion of adversarial examples should be in the range of normal speech.

6. Conclusions

This paper further analyzes why LOGITS NOISE detection can distinguish adversarial instances from normal speech. Furthermore, it defines a quantitative logit distribution difference method. Then, a Logit-Traction attack against the detection is designed. It reduces the accuracy of this detection method by quantifying and increasing the intra-frame differences of the logits of the adversarial examples. Next, to demonstrate that this attack is real and effective, three more quantization methods are designed based on intra-frame mean difference, decision frame variance, and the number of delineation statistics. The experimental results show that a Logits-Traction attack can successfully evade this detection.

Finally, when deploying the detection methods based on logit dispersion, defenders should focus on features with advantages in the model to be protected. For attackers, when the detection method adopted by the target model is unknown, the intra-frame differences of the logits should be improved first.

Author Contributions: Conceptualization, S.H. and M.Y.; Investigation, S.H. and M.Y.; Methodology, S.G.; Project administration, K.X.; Resources, K.X.; Software, S.H.; Supervision, K.X. and S.G.; Validation, S.G.; Writing—original draft, S.H.; Writing—review and editing, S.H. and B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *Comput. Sci.* **2013**, *1312*, 6199.
2. Chang, K.; Huang, P.; Yu, H.; Jin, Y.; Wang, T. Audio Adversarial Examples Generation with Recurrent Neural Networks. In Proceedings of the 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China 13–16 January 2020; pp. 488–493.
3. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA USA, 25–29 June 2006; pp. 369–376.
4. Abdullah, H.; Garcia, W.; Peeters, C.; Traynor, P.; Kevin, R.B.B.; Wilson, J. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–27 February 2019.
5. Samizade, S.; Tan, Z.; Shen, C.; Guan, X. Adversarial Example Detection by Classification for Deep Speech Recognition. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2019; pp. 3102–3106.
6. Rajaratnam, K.; Kalita, J. Noise Flooding for Detecting Audio Adversarial Examples against Automatic Speech Recognition. In Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 6–8 December 2018; pp. 197–201.
7. Kwon, H.; Yoon, H.; Park, K. POSTER: Detecting Audio Adversarial Example through Audio Modification. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2521–2523.
8. Yang, Z.; Li, B.; Chen, P.; Song, D. Characterizing Audio Adversarial Examples Using Temporal Dependency. *arXiv* **2018**, arXiv:1809.10875.
9. Park, N.; Ji, S.; Kim, J. Detecting Audio Adversarial Examples with Logit Noising. In Proceedings of the Annual Computer Security Applications Conference, San Juan, PR, USA, 9–13 December 2021; pp. 586–595.
10. Zhang, H.; Zhou, P.; Yan, Q.; Liu, X. Generating Robust Audio Adversarial Examples with Temporal Dependency. In Proceedings of the International Joint Conferences on Artificial Intelligence Organization, IJCAI-20, Yokohama, Japan, 7–15 January 2021; pp. 3167–3173.
11. Zhang, C.; Benz, P.; Imtiaz, T.; Kweon, I. Understanding Adversarial Examples from the Mutual Influence of Images and Perturbations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14521–14530.
12. Carlini, N.; Wagner, D. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 1–7.
13. Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. Deep Speech: Scaling up end-to-end speech recognition. *arXiv* **2014**, arXiv:1412.5567.
14. Yakura, H.; Sakuma, J. Robust Audio Adversarial Example for a Physical Attack. *arXiv* **2018**, arXiv:1810.11793.
15. Qin, Y.; Carlini, N.; Goodfellow, I.; Cottrell, G.; Raffel, C. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 18–24 July 2019; pp. 5231–5240.
16. Schönherr, L.; Kohls, K.; Zeiler, S.; Holz, T.; Kolossa, D. Adversarial Attacks against Automatic Speech Recognition Systems via Psychoacoustic Hiding. *arXiv* **2018**, arXiv:1808.05665.
17. Taori, R.; Kamsetty, A.; Chu, B.; Vemuri, N. Targeted Adversarial Examples for Black Box Audio Systems. In Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW), Francisco, CA, USA, 19–23 May 2018; pp. 15–20.
18. Godefroid, P.; Khurshid, S. Exploring Very Large State Spaces Using Genetic Algorithms. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Berlin, Heidelberg, 1 January 2002; pp. 266–280.
19. Glasserman, P.; Ho, Y.-C. Gradient Estimation via Perturbation Analysis. *Springer Sci. Bus. Media* **1991**, *33*, 259–260. [CrossRef]
20. Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In Proceedings of the Interspeech-2015, Dresden, Germany, 6–10 September 2015; pp. 3586–3589.

21. Tan, H.; Wang, L.; Zhang, H.; Zhang, J.; Shafiq, M.; Gu, Z. Adversarial Attack and Defense Strategies of Speaker Recognition Systems: A Survey. *Electronics* **2022**, *11*, 2183. [CrossRef]
22. Muda, L.; Begam, M.; Elamvazuthi, I. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv* **2010**, arXiv:1003.4083.
23. Alzantot, M.; Balaji, B.; Srivastava, M. Did you hear that? Adversarial Examples against Automatic Speech Recognition. *arXiv* **2018**, arXiv:1801.00554.
24. Wang, Y.; Xie, L.; Liu, X.; Yin, J.; Zheng, T. Model-Agnostic Adversarial Example Detection through Logit Distribution Learning. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 3617–3621.
25. Aigrain, J.; Detyniecki, M. Detecting Adversarial Examples and Other Misclassifications in Neural Networks by Introspection. *arXiv* **2019**, arXiv:1905.09186.
26. Woodward, A.; Bonnín, C.; Masuda, I.; Varas, D.; Bou-Balust, E.; Riveiro, J.C. Confidence Measures in Encoder-Decoder Models for Speech Recognition. In Proceedings of the Interspeech, Shanghai, China, 25–29 October 2020; pp. 611–615.
27. Bentz, Y.; Merunka, D. Neural networks and the multinomial logit for brand choice modelling: A hybrid approach. *J. Forecast.* **2000**, *19*, 177–200. [CrossRef]
28. Carlini, N.; Wagner, D. Defensive distillation is not robust to adversarial examples. *arXiv* **2016**, arXiv:1607.04311.
29. Raju, A.; Tiwari, G.; Rao, M.; Dheram, P.; Anderson, B.; Zhang, Z.; Bui, B.; Rastrow, A. End-to-end spoken language understanding using rnn-transducer asr. *arXiv* **2021**, arXiv:2106.15919.
30. Shen, J.; Nguyen, P.; Wu, Y.; Chen, Z.; Chen, M.X.; Jia, Y.; Kannan, A.; Sainath, T.; Cao, Y.; Chiu, C.; et al. Lingvo: A Modular and Scalable Framework for Sequence-to-Sequence Modeling. *arXiv* **2019**, arXiv:1902.08295.
31. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (sp), San Jose, CA, USA, 25 May 2017; pp. 39–57.
32. Gao, C.; Cheng, G.; Zhou, J.; Zhang, P.; Yan, Y. Non-autoregressive deliberation-attention based end-to-end ASR. In Proceedings of the 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP), Hong Kong, China, 24–27 January 2021; pp. 1–5.
33. Oh, D.; Park, J.; Kim, J.; Jang, G. Hierarchical phoneme classification for improved speech recognition. *Appl. Sci.* **2021**, *11*, 428. [CrossRef]
34. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
35. Jang, U.; Wu, X.; Jha, S. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, 4–8 December 2017; pp. 262–277.
36. Chen, X.; Li, S.; Huang, H. Adversarial Attack and Defense on Deep Neural Network-Based Voice Processing Systems: An Overview. *Appl. Sci.* **2021**, *11*, 8450. [CrossRef]
37. Qin, Y.; Carlini, N.; Cottrell, G.; Goodfellow, I.; Raffel, C. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. Available online: https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/adversarial_asr (accessed on 20 May 2022).
38. Park, N.; Ji, S.; Kim, J. Detecting Audio Adversarial Examples with Logit Noising. Available online: https://github.com/namgyupark22/_Detecting_Audio_Adversarial_Examples_with_Logit_Noising (accessed on 20 May 2022).
39. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5206–5210.
40. Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F.M.; Weber, G. Common voice: A massively-multilingual speech corpus. *arXiv* **2019**, arXiv:1912.06670.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.