Detecting and Defending Against Adversarial Attacks in Speech-to-Text Applications via Diffusion Models

Astrid Holm Filtenborg Kitchen, Marie Saugstrup Jensen, Mikkel S. Lundsgaard Brøndt, and Nikolai Lund Kühne
7th Semester of Mathematical Engineering at Aalborg University
{akitch20, marije19, mbrand20, nkahne20}@student.aau.dk

Abstract—**Machine learning models such as speech-to-text applications are vulnerable to adversarial attacks. This paper focuses on defending against non-adaptive targeted black- and white-box attacks on speech signals for speech-to-text applications. Denoising diffusion probabilistic models are presented as they are used to defend against targeted attacks. Existing work in the speech domain have used diffusion models to purify adversarial examples and obtained state-of-the-art results on keywords from the Google Speech Commands dataset with a keyword spotter as classifier. In this paper, these results were extended by testing diffusion based adversarial purification with different amounts of diffusion steps on full sentences. Specifically, 300 short, 300 medium, and 300 long sentences from the Mozilla Common Voice dataset were perturbed through the state-of-the-art Carlini & Wagner white box attacking method. With this, it was shown that diffusion models can be used for defending against adversarial attacks and purifying full sentence adversarial examples. Following a series of comprehensive experiments, we have achieved 100% success rate when defending against adversarial attacks on sentences using two forward diffusion steps. Finally, we leveraged a pre-trained diffusion model to introduce a novel approach for detecting adversarial attacks that do not require model training. We could detect if an input was adversarial or benign with at least 86% accuracy, and adversarial examples were correctly detected at least 94% of the time.**

Code is publicly available at https://github.com/Kyhne/Detecting-and-Defending-Against-Adversarial-Attacks.

Index Terms—**Diffusion models, adversarial attacks, speech recognition, deep learning.**

I. Introduction

Machine learning models are vulnerable to adversarial attacks, which add a small perturbation to the original input such that a machine learning model returns a wrong output, while being imperceptible to humans. This paper focuses on adversarial attacks in the audio domain, where an attack can be embedded into speech signals such that speech-to-text applications transcribe different words than what has been said. This is an issue, since smart devices using voice commands can be attacked with hidden voice commands registered only by the device. Thus, by playing a video with a hidden voice command, it is possible to make a smart device visit a malicious web page causing a drive-by download [1]. This motivates detecting and defending against adversarial attacks.

There are two different categories of adversarial attacking methods: black-box (BB) attacks and white-box (WB) attacks. When the layers and parameters of the machine learning model are unknown to the attacker, the adversarial attack is known as a BB attack. On the contrary, if all layers and parameters are fully accessible, then the adversarial attack is known as a WB attack. Furthermore, these two methods are usually divided into two main categories: targeted and non-targeted attacks. The goal of targeted attacks is to make the machine learning model return a particular target output different from the expected output. For non-targeted attacks, it is sufficient that the output is different from the expected output. This paper focuses on defending against non-adaptive targeted BB and WB attacks on speech signals for speech-to-text applications. In addition, adversarial detection of WB attacks on sentences is examined.

In [2], adversarial examples in speech recognition applications are detected through classification scores, but their proposed method cannot be applied, when the classification scores assigned by the target model are unknown. In [3], a method using a convolutional neural network structure with small kernels was proposed in order to detect small perturbations in adversarial examples in speech recognition applications. They were able to train a machine learning model to accurately detect adversarial examples generated from known attacking methods.

To defend against adversarial attacks, [4], [5] have proposed an approach that utilises the temporal dependency property of audio, where time domain and frequency domain transformations are applied to the adversarial example to improve robustness. These solutions consist of temporal smoothing, down sampling, and low-pass filtering. These approaches do not require training, which is the case for the adversarial training based defence proposed in [6], where adversarial perturbations are added to the training stage. This is the most effective defence according to [7], however, the model is still vulnerable to adversarial attacks that are different to those used in the training process.

In [7], an adversarial purification-based defence pipeline called AudioPure is proposed for the audio domain, which uses diffusion models to remove the adversarial perturbations. Two different diffusion models, DiffWave [8] and DiffSpec [9], are used as waveform and spectrogram purifiers in AudioPure, respectively, where DiffWave consistently outperforms DiffSpec. DiffWave is based on denoising diffusion probabilistic models (DDPMs) [10], which are probabilistic generative models that gradually perturbs data to noise and subsequently learn how to convert noise back to data [11]. AudioPure is tested on two WB attacks and one BB attack, where the digits 0-9 from the Google Speech Commands (GSC) dataset [12] are used as utterances. Moreover, they only have results for a fixed amount of forward diffusion steps. Compared to six other both standard and cutting-edge defence methods, AudioPure was shown to be the most effective in defending against the WB attacks and is the second best for the BB attack.

This paper aims to verify the results from [7] by applying the same defensive purifier on different adversarial attacks and speech-to-text applications than used in their paper. We extend these results by varying the amount of forward diffusion steps to examine what impact it has on the defence against adversarial attacks. This work is further extended by defending against adversarial attacks on sentences. A novel approach for detecting unknown adversarial attacks on sentences is proposed by leveraging the denoising capabilities in diffusion models, through comparing the classifier output on clean- and purified-speech signals. Finally, our results from the detection experiment and model limitations are compared to the detection results in [3].

II. Adversarial Attacking Methods

Two adversarial attacking methods have been chosen: the Alzantot [13] attacking method and the Carlini & Wagner (C&W) [14] attacking method.

The Alzantot attacking method is a BB attack that uses a genetic algorithm, which is a gradient-free optimisation method. It has been chosen since it achieves a success rate of 100% on untargeted attacks and approximately 87% on targeted attacks [13].

The C&W attacking method is a WB attack, which has been chosen as it is one of the most successful WB attacking methods for audio, achieving 100% success rate in their experiments. The adversarial perturbation is optimised by using iterative optimisation to solve the optimisation problem

$$\underset{\delta}{\text{minimise}} \quad \|\delta\|_2^2 + c \cdot \ell(x+\delta, t)$$
$$\text{such that } dB_x(\delta) \leq \tau,$$

where δ is the added adversarial perturbation, $\ell(\cdot)$ is the connectionist temporal classification loss function [15], x is the original example, and t is the desired target phrase. The parameter c is being updated to ensure $x+\delta$ becomes adversarial while remaining close to x. The constant τ is sufficiently large and when a partial solution δ^* is obtained, τ is reduced until no solution can be found. Lastly, $dB_x(\delta) = dB(\delta) - dB(x)$ is the decibel level of the perturbation δ compared to x, where $dB(x) = \max_i 20 \cdot \log_{10}(x_i)$.

Figure 1 illustrates three spectrograms, which show the difference between the original example and the adversarial examples. It is seen that the C&W attacking method makes less changes in the speech region compared to the Alzantot attacking method, and the two attacking methods make different perturbations in the non-speech area.

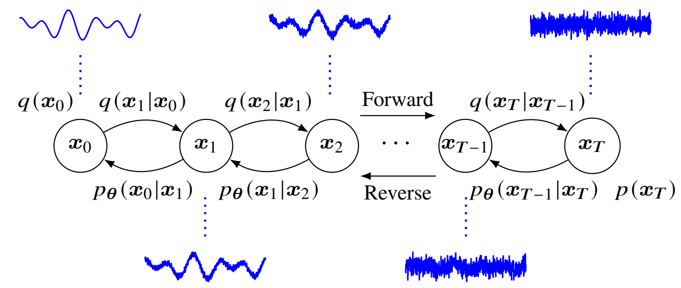

Figure 2. The forward and reverse diffusion process in DDPMs. The reverse diffusion process gradually converts the white noise signal to recover clean data through a Markov chain $p_\theta(x_{t-1}|x_t)$. Inspired by [8].

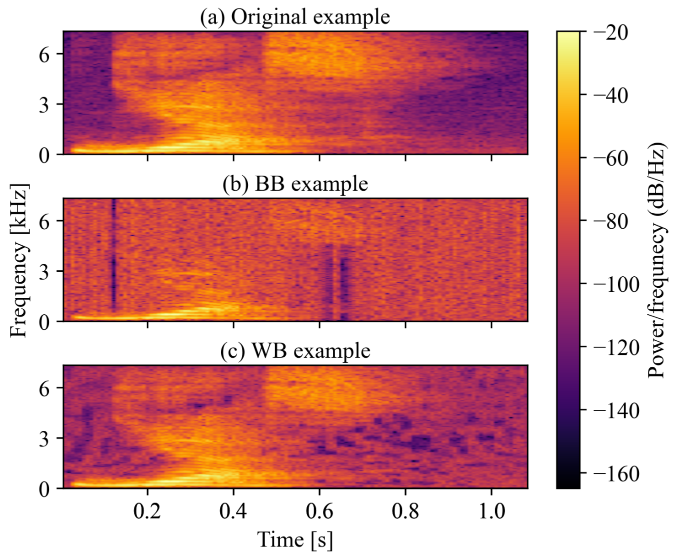

Figure 1. Spectrogram (a) is the original example where the original sentence is “yes”. Spectrograms (b) and (c) are adversarial examples generated by the Alzantot and C&W attacking methods, respectively, using the same target phrase “right”. Inspired by [3].

III. Diffusion Models

A. *Denoising Diffusion Probabilistic Models*

DDPMs [10] are one of the most extensively used diffusion models. A DDPM consists of a forward diffusion process and a reverse diffusion process, which are defined by Markov chains. The forward diffusion process gradually adds noise to the input data until the distribution of the noisy data is approximately equal to a standard Gaussian distribution. The reverse diffusion process is parameterised by a deep neural network that takes the approximately standard Gaussian noise as input and gradually denoises the noisy data to recover clean data. This process can be seen in Figure 2.

Formally, let $\boldsymbol{x}_0 \in \mathbb{R}^d$ and $q(\boldsymbol{x}_0)$ be an unknown data distribution. Then for each data point $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, a forward Markov chain is formed such that $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}), 1 \leq t \leq T$ based on a predetermined noise variance schedule $0 < \beta_1, \beta_2, \ldots, \beta_T < 1$. To elaborate, $\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Using the reparameterisation trick with $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{i=0}^t \alpha_i$ results in $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\boldsymbol{I})$. The predetermined noise variance schedule is chosen such that $\overline{\alpha}_T \approx 0$, which results in $q(\boldsymbol{x}_T|\boldsymbol{x}_0) \approx \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and thus $q(\boldsymbol{x}_T) \approx \mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \boldsymbol{I})$. In fact, as $T \to \infty$ the resulting distribution of $\boldsymbol{x}_T$ is isotropic Gaussian, since the result of this limit is a variance preserving stochastic differential equation [16]. As both $q(\boldsymbol{x}_0|\boldsymbol{x}_T)$ and $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ are intractable, the denoising process $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is approximated by a learnable Markov chain $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}|\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t))$ given a prior distribution $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \boldsymbol{I})$. Specifically,

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t)\right), \quad \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t) = \tilde{\beta}_t\boldsymbol{I},$$

where $\hat{\epsilon}_{\theta} : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^d$ is a deep neural network predicting the noise $\epsilon_0 \sim \mathcal{N}(\epsilon|\mathbf{0}, \boldsymbol{I})$ in $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0$, and $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$. By [7], $\boldsymbol{\Sigma}_{\theta}(\boldsymbol{x}_t, t) = \tilde{\beta}_t \boldsymbol{I}$ as it yields the best results. The model is trained by maximising its variational lower bound (VLB):

$$\begin{aligned} L_{\text{VLB}} &= \mathbb{E}_{q(\boldsymbol{x}_0)q(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log\left(\frac{q(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T|\boldsymbol{x}_0)}{p_{\theta}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T)}\right)\right] \\ &\geq -\mathbb{E}_{q(\boldsymbol{x}_0)}\left[\log(p_{\theta}(\boldsymbol{x}_0))\right], \end{aligned}$$

since the likelihood $p_{\theta}(\boldsymbol{x}_0) = \int p_{\theta}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T)\,\mathrm{d}(\boldsymbol{x}_1\ldots,\boldsymbol{x}_T)$ is intractable. This results in the following objective function:

$$L_{t-1} = \\ \mathbb{E}_{\boldsymbol{x}_0,\epsilon_0}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t(1-\overline{\alpha}_t)\alpha_t}\left(\left\|\epsilon_0 - \hat{\epsilon}_{\theta}\left(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0, t\right)\right\|_2^2\right)\right],$$

where $2 \leq t \leq T$. In [10], it was found that a simplified training objective yields better samples. Let $t \sim \text{unif}(1,T)$, then

$$L_{\text{simple}} = \mathbb{E}_{t,\boldsymbol{x}_0,\epsilon_0}\left[\left\|\epsilon_0 - \hat{\epsilon}_{\theta}\left(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0, t\right)\right\|_2^2\right].$$

The training process is described in Algorithm 1.

Algorithm 1: DDPM Training

```
while not converged do
    Sample $x_0 \sim q(x_0), t \sim \text{unif}(1,T)$ and
    $\epsilon_0 \sim \mathcal{N}(0, I)$
    Calculate
    $\theta_\nabla = \nabla_\theta \left\|\epsilon_0 - \hat{\epsilon}_\theta\left(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0, t\right)\right\|_2^2$
    Take gradient descent step $\theta := \theta + \gamma\theta_\nabla$
end
```

[10, Algorithm 1]

The algorithm for sampling from DDPMs is presented in Algorithm 2, since DiffWave will be used in an unconditional generation task based on raw waveform only to purify adversarial examples. The algorithm can be interpreted as sampling from the Markov chain defined by the reverse chain $p_{\theta}(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_T) = p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\theta}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, where each step $p_{\theta}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ has mean $\boldsymbol{\mu}_{\theta}(\boldsymbol{x}_t, t)$ and covariance $\tilde{\beta}_t\boldsymbol{I}$.

Algorithm 2: DDPM Sampling

```
Sample $x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 2$ do
    Sample $\epsilon_0 \sim \mathcal{N}(0, I)$
    Calculate
    $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\hat{\epsilon}_\theta(x_t, t)\right) + \sqrt{\tilde{\beta}_t}\epsilon_0$
end
return $x_0 = \frac{1}{\sqrt{\alpha_1}}\left(x_1 - \frac{1-\alpha_1}{\sqrt{1-\overline{\alpha}_1}}\hat{\epsilon}_\theta(x_1, 1)\right)$
```

[10, Algorithm 2]

The adversarial purification-based defence used in this paper, denoted **Purifier**, is based on the plug-and-play method AudioPure [7]. AudioPure is an implementation of a DDPM, which is based on a pre-trained DiffWave [8] model. The **Purifier** will be applied to a waveform input before being fed into a pre-trained classifier to defend against adversarial attacks.

B. Formulation of Defence

Let $\boldsymbol{x}_{\text{adv}} \in \mathbb{R}^d$ be the adversarial waveform input. To override the adversarial perturbations, the **Purifier** starts by adding noise to $\boldsymbol{x}_{\text{adv}}$ via the forward diffusion process. Through the reverse diffusion process, the goal is to remove the adversarial perturbations in order to recover the clean sample which is fed into the classifier. This process is portrayed in the grey box in Figure 3.
The pre-trained DDPM DiffWave consists of a 36-layer residual neural network and uses a feed-forward and bidirectional dilated convolution architecture. The kernel size is 3, the dilution cycle is $[1, 2, \ldots, 2048]$, and the amount of residual channels is $C = 256$. Furthermore, the number of forward diffusion steps is $T = 200$, and the noise schedule is linearly spaced for $\beta_t \in [0.0001, 0.02]$. From the T diffusion steps, the first $n^\star \in [1, 2, \ldots, T]$ steps are chosen in order to avoid adding too much noise before running the reverse process, since adding too much noise not only overrides the adversarial perturbations, but also destroys information from the original signal. Then, the **Purifier** takes $\boldsymbol{x}_{\text{adv}}$ and $n^\star$ as inputs, and returns the purified audio, where this operation is denoted as a function **Purifier** : $\mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^d$. Subsequently, the classifier $F : \mathbb{R}^d \to \mathbb{R}^c$ is applied to the purified audio, where $c \in \mathbb{N}_0$. Finally, the purifier and classifier are combined into a defended speech system $\mathbf{SS} : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^c$ given by

$$\mathbf{SS}\left(\boldsymbol{x}_{\text{adv}}, n^\star\right) = F\left(\mathbf{Purifier}\left(\boldsymbol{x}_{\text{adv}}, n^\star\right)\right).$$

The entire speech system is shown in Figure 3. As opposed to [7], we do not use the Wave2Mel function, which applies a short-time Fourier transformation on the time domain waveform to get a linear-scale spectrogram after which the frequency band is rescaled to the Mel-scale. This is not necessary, since Deep Speech is a waveform classifier.

IV. Experiments

A. Methodology for Defending Against Adversarial Attacks

The detailed experimental settings for the BB and WB attacks as well as the adversarial purification are presented in this section.

Datasets. A subset of the Google Speech Commands dataset v0.01 [12] is used, which consists of 85,511 training utterances, 10,102 validation utterances, and 4,890 test utterances. From this, 10 keywords are chosen: “Yes”, “No”, “Up”, “Down”, “Left”, “Right”, “On”, “Off”, “Stop”, and “Go”. The first 20 utterances of each keyword in the dataset are chosen, resulting in 200 different audio signals. These 200 audio signals comprise the clean-speech inputs to the classifier. For the BB attack, 180 adversarial examples are generated for each keyword by using mutual targeting, where the other nine keywords are targets for the attacked keyword, resulting in a total of 1800 adversarial examples. This is repeated for the WB attacks.
The experiment for WB attacks is extended by attacking and

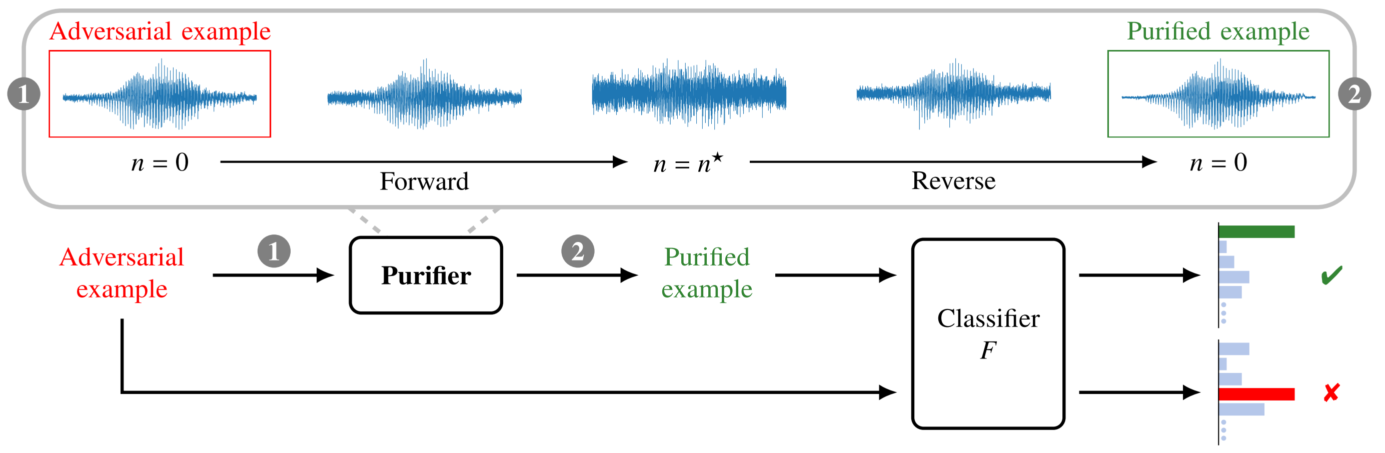

Figure 3. The architecture of the whole speech system protected by the **Purifier**. The **Purifier** first adds noise to the adversarial example via the forward diffusion process and then runs the reverse process to recover the purified example. Nextly, the purified example is fed into the classifier to get predictions. Without the **Purifier**, the adversarial example is fed into the classifier directly. Inspired by [7].

defending entire sentences from the Mozilla Common Voice (MCV) dataset [17]. For this, 300 short, 300 medium, and 300 long sentences with length 1-2, 3-4 and 6-7 seconds, respectively, are chosen from the dataset and the same target is chosen for all attacked sentences of the same length:

- **Short:** "open all doors".
- **Medium:** "turn off internet connection".
- **Long:** "i need a reservation for sixteen people at the seafood restaurant down the street".

Only audio files that contain more than 68% speech have been chosen using the open-source robust Voice Activity Detection (rVAD) algorithm [18].

Models. The unconditional version of DiffWave [8] with the officially provided pre-trained checkpoints is utilised as the defensive waveform purifier. For the classifier, a keyword spotter [19] is used for the BB attack. For the WB attacks, the end-to-end Deep Speech system [20] is used as a classifier. Note that the classifier for the BB attack only has 12 different classification targets: the 10 keywords, "silence", and "unknown". Deep Speech can, however, output a sequence of characters from the English alphabet, thus allowing for more classification outputs.

Attacks. For the BB attack, the Alzantot [13] attacking method is used with a noise budget of 10 and 500 iterations. For the WB attack on the GSC dataset, the C&W [14] method is used, where the maximum amount of iterations is 1000 and the learning rate for the gradient descent optimisation is 100. For the WB attack on the MCV dataset, the maximum amount of iterations is 5000 and the learning rate is 10 just as in [14].

Evaluation. The success of the adversarial attacks is evaluated as the percentage of times the classifier returns exactly the target.
For keywords, the purification is evaluated as the percentage of times the purified keyword is classified exactly as the spoken keyword.
For sentences, the purification is evaluated by calculating the character error rate (CER) between the spoken sentence and the classifier output. Here, $1 - \text{CER}$ is returned to represent the success rate. The CER is used, since otherwise the whole sentence would be deemed incorrect if only a single character was transcribed wrongly.

A summary of the preceding information can be seen in Table I for an overview.

Table I. Overview of datasets, utterances, attacking methods, and classifiers.

Dataset	Google Speech Commands v.0.01		Mozilla Common Voice
Utterances	Words		Sentences
Attack	Black-box	White-box	White-box
Classifier	Keyword spotter	Deep Speech	Deep Speech

B. Methodology for Adversarial Detection

Adversarial detection is only performed on sentences using the same 900 sentences, WB attacking method, Deep Speech classifier, and purifier as detailed in subsection IV-A. Attacking all 900 sentences yields 300 adversarial and 300 benign sentences of each sentence length resulting in 1800 sentences. For each sentence length, the data is split into two parts containing 10% and 90% both containing 50% adversarial and 50% benign sentences.
The detection method is a binary classification method as it can only return the labels adversarial or benign. To detect an adversarial attack on an input sentence $\boldsymbol{x}_{\text{in}}$, the CER between the clean-speech classifier output $F(\boldsymbol{x}_{\text{in}})$ and purified-speech classifier output $\mathbf{SS}(\boldsymbol{x}_{\text{in}}, n^\star)$ is calculated. If $\text{CER}(\mathbf{SS}(\boldsymbol{x}_{\text{in}}, n^\star), F(\boldsymbol{x}_{\text{in}})) > \Omega$, then $\boldsymbol{x}_{in}$ is classified as adversarial, otherwise $\boldsymbol{x}_{\text{in}}$ is classified as benign. The pre-determined threshold Ω is found based on the first 10% data. For each sentence length and forward diffusion step $n^\star \in \{1, 2, 3, 4, 5\}$, a grid search is performed to find the Ω that maximises the accuracy of the adversarial detection. The detection method is evaluated using a confusion matrix as well as a receiver operating characteristic (ROC) curve. In the confusion matrix, a positive or negative outcome refers to an input being classified as adversarial or benign, respectively. These results are based on the remaining 90% of data.

Table II. Accuracy of the classifier in percentage. C-KS and C-DS stand for clean audio through the keyword spotter and Deep Speech system, respectively, and S, M, L stands for short, medium and long, respectively.

Defence	Keywords				Sentences					
	C-KS	BB	C-DS	WB	C-DS-S	WB-S	C-DS-M	WB-M	C-DS-L	WB-L
None	**93.00**	9.83	**65.00**	1.44	**78.00**	19.41	**86.01**	20.58	**86.07**	24.67
Purifier $n^\star = 1$	92.00	64.22	53.50	**49.89**	71.88	41.97	76.94	34.70	73.62	36.00
Purifier $n^\star = 2$	89.00	73.17	48.50	42.50	64.67	45.12	67.47	40.23	65.50	40.29
Purifier $n^\star = 3$	88.50	76.44	46.50	37.67	59.34	46.40	61.02	44.50	57.76	44.07
Purifier $n^\star = 4$	82.00	77.06	37.50	35.61	54.45	**46.70**	55.66	**46.16**	52.60	**45.31**
Purifier $n^\star = 5$	80.50	**77.33**	30.50	31.50	49.86	45.73	51.02	45.17	47.12	44.34

V. Results

Table II and Table III show the accuracy of the classifier and the success rate of the adversarial attacks, respectively. Table IV shows the results from the detection experiment.

A. Purification on Keywords

Generally, the results from [7] have been reproduced with different adversarial attacks. In Table II, it is seen that for clean-speech inputs, the accuracy of the keyword spotter and the Deep Speech system decrease as $n^\star$ increases. For the BB attack, the highest accuracy of the keyword spotter is achieved when $n^\star = 5$ whereas for the WB attack, the Deep Speech system reaches the highest accuracy when $n^\star = 1$. In Table III, it is seen that the success rates of the BB and WB attacks drop to 1.28%, when $n^\star = 4$ and $n^\star = 5$, respectively.

B. Purification on Sentences

The accuracy in Table II drops 58.59 pp, 65.43 pp, and 61.40 pp when comparing the clean and attacked sentences without defence on the short, medium and long sentences, respectively. For attacked sentences, the accuracy of the classifier is the highest when $n^\star = 4$ across all sentence lengths.

The success rate of the adversarial attacks in Table III reach 0.00% when $n^\star = 1$ for medium sentences and $n^\star = 2$ for short and long sentences, meaning the adversarial attack is completely defended against every time.

Table III. Success rate of the adversarial attacks in percentage, where S, M, L stands for short, medium and long, respectively.

Defence	Keywords		Sentences		
	BB	WB	WB-S	WB-M	WB-L
None	85.61	55.11	71.33	90.67	94.33
Purifier $n^\star = 1$	6.94	3.22	1.67	**0.00**	0.33
Purifier $n^\star = 2$	3.28	2.44	**0.00**	0.00	**0.00**
Purifier $n^\star = 3$	2.06	2.22	0.00	0.00	0.00
Purifier $n^\star = 4$	**1.28**	1.72	0.00	0.00	0.00
Purifier $n^\star = 5$	1.72	**1.28**	0.00	0.00	0.00

C. Adversarial Detection

Based on the grid search, the optimal threshold was found to be 0.57 with forward diffusion steps $n^\star = 2$ for all sentence lengths. With this threshold, the accuracy is at least 86% as seen in Table IV. The percentages of true positive classifications are all above 94%, and the percentages of false negative classifications are all below 6%, meaning that the WB attacks are detected in the majority of the cases. Furthermore, the percentage of false positive classifications are all higher than the percentage of false negative classifications meaning that more benign inputs are classified as adversarial than adversarial inputs are classified as benign.

Table IV. Results from the detection experiment given the optimal threshold 0.57 and forward diffusion steps $n^\star = 2$. TN stands for true negative, FN for false negative, FP for false positive, and TP for true positive.

Length		Short		Medium		Long	
TN	FP	0.75	0.25	**0.82**	**0.18**	0.78	0.22
FN	TP	**0.02**	**0.98**	0.06	0.94	0.04	0.96
Accuracy		0.86		**0.88**		0.87	
Precision		0.80		**0.84**		0.81	
Sensitivity		**0.98**		0.94		0.96	
Specificity		0.75		**0.82**		0.78	
AUC		**0.91**		0.89		0.88	

Figure 4 visualises the results from Table IV through a ROC curve. The highest value of the area under the curve (AUC) is for short sentences with 0.91 and in general the ROC curves look healthy.

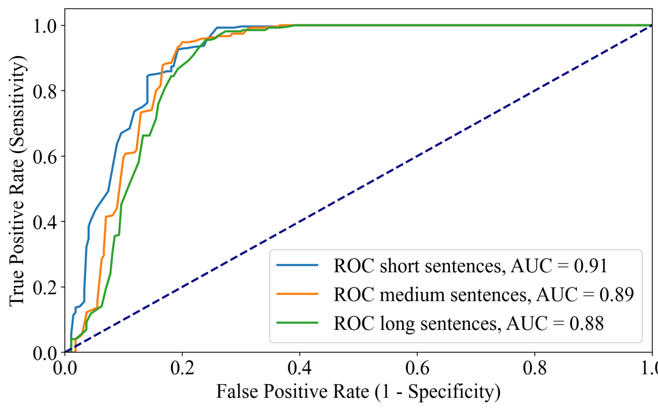

Figure 4. ROC curves of the detection method for $n^\star = 2$ and threshold 0.57.

VI. Discussion

We will not discuss the results of the purification on keywords, since the main focus in this paper is detecting and defending against adversarial attacks on sentences.

Without any defence, it is seen from Table III that the success rates of the WB attack on sentences are high, where it is 94.33%

for WB-L. Consequently, the accuracy of the Deep Speech system without any defence is expected to be low. It is however seen in Table II that this accuracy is 24.67% for WB-L. This is a consequence of using the CER to evaluate the accuracy of the classifier. In fact, calculating $1 - \mathrm{CER}$ between the spoken sentences and the chosen targets to simulate 100% success rate of the WB attack, results in 19.75%, 20.69%, and 24.55% success rate for short, medium, and long sentences, respectively. These numbers are almost equivalent to the corresponding results for sentences without any defence in Table II.

From Table III, the WB attack is completely defended against when $n^\star = 2$ for all sentence lengths. For the attacked sentences in Table II, the best classifier accuracies for all sentence lengths are obtained when $n^\star = 4$. Meaning, the **Purifier** can be used to purify adversarial examples and completely defend against WB attacks. However, the clean signal gets misclassified more often when $n^\star$ increases, as an increasing amount of the original waveform is removed. This is seen in Table II, where the accuracy of the classifier drops more than 28 pp in all three cases going from no defence to the **Purifier** with $n^\star = 5$.

In order to avoid classifying purified-speech signals, the **Purifier** is instead used to detect adversarial attacks. From Table IV it is seen that most of the adversarial attacks are classified as adversarial. Consequently, the percentages of false negative classifications are low which is desirable, as we do not want any adversarial examples to go undetected. The percentage of false positive classifications is between 18% and 25%, which is desired to be lower. However, we would much rather misclassify a clean signal than an attacked signal. With these results, our proposed approach can be used for detecting adversarial attacks.

The method presented in [3] achieves better detection results when testing on adversarial attacks introduced in the training phase. In the case of unknown adversarial attacks, the performance of their detection method degrades dramatically. Our approach does not require training and should therefore not be limited to certain adversarial attacks, although it has only been tested on one WB attack unknown to the model. However, as noted in [7], the computational cost of using diffusion models is high, limiting the use of our proposed detection method in time-sensitive applications.

VII. Conclusion

In this paper, we leverage a pre-trained diffusion model to defend against adversarial attacks on full sentences. The defence has 100% success rate when using two forward diffusion steps in the **Purifier**. Comprehensive experiments indicate that the use of the **Purifier** on clean-speech signals is detrimental to classification performance. Hence, we introduced a novel approach for detecting unknown adversarial attacks on sentences that does not require any training. Experiments have shown that we achieve at least 86% accuracy on the detection, where an adversarial input is correctly detected at least 94% of the time.

Acknowledgement

We thank our supervisors Zheng-Hua Tan, Christophe Biscio, and Martin Gonzalez for guidance and feedback. We also thank the authors of the Mozilla Common Voice dataset and the Google Speech Commands dataset for making their datasets publicly available. Furthermore, we thank Nicholas Carlini and David Wagner, and Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava for providing code used and we thank Baidou Deep Speech for making Deep Speech publicly available.

References

[1] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 513–530.

[2] H. Kwon and S.-H. Nam, "Audio adversarial detection through classification score on speech recognition systems," *Computers & Security*, vol. 126, p. 103061, 2023.

[3] S. Samizade, Z.-H. Tan, C. Shen, and X. Guan, "Adversarial example detection by classification for deep speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3102–3106.

[4] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," *arXiv preprint arXiv:1809.10875*, 2018.

[5] K. Rajaratnam, B. Alshemali, and J. Kalita, "Speech coding and audio preprocessing for mitigating and detecting audio adversarial examples on automatic speech recognition," 2018. [Online]. Available: http://cs.uccs.edu/~jkalita/work/reu/REU2018/07Rajaratnam.pdf

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] S. Wu, J. Wang, W. Ping, W. Nie, and C. Xiao, "Defending against adversarial audio via diffusion model," *arXiv preprint arXiv:2303.01507*, 2023.

[8] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2021.

[9] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[11] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[12] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[13] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[14] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine learning*, 2006, pp. 369–376.

[16] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2021.

[17] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2020.

[18] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 798–807, 2010.

[19] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv preprint arXiv:1703.05390*, 2017.

[20] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.