Self-Supervised Fine-Tuning of Automatic Speech Recognition Systems against Signal Processing Attacks

Oshan Jayawardena
oshanjayawardanav100@gmail.com
University of Moratuwa
Moratuwa, Sri Lanka

Dilmi Caldera*
diljc98@gmail.com
University of Moratuwa
Moratuwa, Sri Lanka

Sandani Jayawardena*
sandaninavanjana@gmail.com
University of Moratuwa
Moratuwa, Sri Lanka

Avishka Sandeepa*
cavishkasandeepa@gmail.com
University of Moratuwa
Moratuwa, Sri Lanka

Vincent Bindschaedler
vbindsch@cise.ufl.edu
University of Florida
Florida, USA

Subodha Charles
scharles@uom.lk
University of Moratuwa
Moratuwa, Sri Lanka

ABSTRACT

Automatic Speech Recognition (ASR) systems take audio signals as inputs and output the corresponding text transcriptions. The text is then used to execute commands and perform searches in several application domains, including security-critical applications such as smartphone assistants, smart home assistants, and self-driving car assistants. Signal processing attacks are one of the most recent types of attacks designed to fool ASR models. Signal processing attacks exploit the feature extraction stage of the ASR pipeline and add perturbations to the audio. These attacks are capable of generating wrong transcriptions of the audio signals even though the attacked audio sounds similar to the original audio. Existing defences for adversarial attacks are neural networks that act as a filter to remove attacks from audio waveforms. The heuristic-based training objective function used in training these filter networks has a negative impact on the performance. Also, there is a disconnect between the training objective function and the application objective function. We address these problems and propose a novel self-supervised fine-tuning algorithm to make existing ASR models robust to adversarial attacks. We do extensive experimentation on our method against signal processing attacks across four different scenarios, and in three out of four scenarios, our method exhibits the best results.

CCS CONCEPTS

• **Computing methodologies** → *Speech recognition.*

KEYWORDS

adversarial attacks, automatic speech recognition, self-supervised learning

*All three authors contributed equally to this research.

ASIA CCS '24, July 1–5, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0482-6/24/07
https://doi.org/10.1145/3634737.3645013

ACM Reference Format:
Oshan Jayawardena, Dilmi Caldera, Sandani Jayawardena, Avishka Sandeepa, Vincent Bindschaedler, and Subodha Charles. 2024. Self-Supervised Fine-Tuning of Automatic Speech Recognition Systems against Signal Processing Attacks. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 1–5, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3634737.3645013

1 INTRODUCTION

ASR systems are widely used as a new form of accessibility and new products are starting to incorporate them into their devices. An application of ASR — speech-to-text (STT) models is essential in enabling voice assistants (e.g., Siri, Google Assistant, Alexa) and real-time captioning services. STT models struggled in robustness, but most of the robustness challenges have been addressed recently due to the availability of large datasets and improved hardware. A prominent example of this is OpenAI's Whisper STT model [26]. However, the security of STT models is still being debated. Even the commercially available, state-of-the-art STT models are vulnerable to attacks. If voice commands are to be used in security-critical applications such as smart home assistants, smartphones, electric vehicles, etc., ASR systems must be resilient to adversarial attacks. Adversarial attacks are attacks that perturb the original audio with the intention of misdirecting the STT model to produce an incorrect output.

Signal processing attacks are a class of adversarial attacks that exploit the feature extraction stage of the ASR pipeline [3, 6]. Even though the signal processing attack does not directly target the inference component, it forces the output to be malicious. It can be an addition or subtraction of a frequency component, a clipping of intensities of audio samples which are higher than a threshold, or an addition of Gaussian noise. Although these perturbations could mislead the STT models, they are imperceptible to the human ear. Figure 1 shows an illustrative example of a signal processing attack. In the illustrative example, the voice command given by the user - "Call my barber" is intentionally perturbed by an attacker. Therefore the STT model gets the adversarial audio sample as input, which outputs a completely different text - "Close my bank account". However, when a human listens to the adversarial audio sample, the human perceives it as "Call my barber". Therefore, the attack is deemed to be imperceptible to the human ear.

In contrast to signal processing attacks, there are two other main types of attacks — (i) optimization attacks [2, 10, 13, 15, 20, 25,

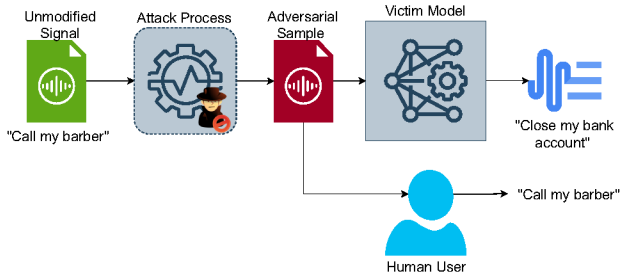

Figure 1: Illustrative example of a signal processing attack on an ASR system.

27, 34, 35], and (ii) gradient-free attacks [10, 14, 30] explored in previous work. We focus on signal processing attacks due to their high transferability and near real-time performance in real-world ASR systems [6]. The behaviour of signal processing attacks makes them faster, less model-dependent, and more query efficient. Signal processing attacks are one of the most recent types of attacks. To our surprise, most of the state-of-the-art commercial ASR systems are vulnerable to these attacks, even after several defences have been proposed.

The goal of a defence against such attacks must either be to detect attacked audio samples and discard them before feeding into the STT model or to change the attacked audio sample such that the changes made during the attack are removed or minimized. In security-critical applications, it is best to discard any audio samples that are being flagged as attacked. However, some applications can tolerate transcriptions with small errors. Some examples are video auto-captioning, telephony surveillance and flagging and audio journals. The main focus of this paper is the latter type of applications. Therefore, we propose a defence to reduce the Word Error Rate (WER) caused by attacked audio samples.

In this work, we critically analyze existing defences for signal processing attacks and outline what factors affect the robustness of defences. We show how the disjoint nature between the training objective and the application objective impacts the performance of the existing methods. Previous work approached this problem based on heuristics [11]. The existing methods train a model that acts as a filter to remove attacks from audio. During the training, an objective function based on heuristics is used. The objective functions try to minimize the distance between the original waveform and the adversarial waveform instead of considering the application objective of minimizing the WER [11]. In fact, it is difficult to connect these two objectives since reconstructing the audio should be done in a continuous space and minimizing the WER should be done in a discrete space. Connecting these two objectives will, in most cases, lead to a non-differentiable loss function.

Inspired by the work of Zhao et al. [37], we propose to do the reconstruction in a latent space that is between the continuous and discrete spaces. While Zhao et al.'s work showed that using a latent space works well in generating natural adversarial examples in both image and text domains, we use a latent space to come up with an improved defence against signal processing attacks. We introduce a novel self-supervised fine-tuning algorithm which incorporates the latent space of the STT model to make STT models resilient against signal processing attacks. Our defence only requires fine-tuning the encoder of STT models. This saves both time and cost compared to fine-tuning an entire STT model end-to-end. We demonstrate our algorithm by fine-tuning the Whisper STT model [26]. We call the improved Whisper, the "Robust Whisper".

It is important to be mindful of what happens to benign audio samples (audio samples that are not attacked) while we try to defend against attacked audio samples. We specifically design the loss function of the fine-tuning algorithm to reduce the damage on benign samples as much as possible because, ideally, benign audio samples must be fed into the STT model unchanged. However, to further improve performance on benign samples, we propose to complement our approach with an attack detection model that detects whether a given audio sample is attacked or not.

By combining the fine-tuning algorithm and the detection model, we come up with an end-to-end solution as the defence against signal processing attacks. The audio samples that are flagged as attacked from the attack detection model are fed into Robust Whisper and the benign samples are sent to the original Whisper model (Vanilla Whisper). We call the combination of Robust Whisper and the detector, "Regularized Robust Whisper". Our defence, Regularized Robust Whisper, is illustrated in Figure 2.

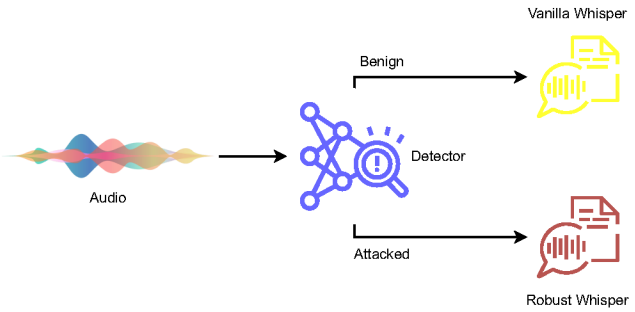

Figure 2: Regularized Robust Whisper. The figure illustrates the pipeline of the end-to-end defence mechanism. The audio waveforms are first classified as benign or attacked using the attack detection model. Only the waveforms flagged as attacked are sent through the fine-tuned Robust Whisper. The waveforms flagged as benign are sent through the original Whisper model (Vanilla Whisper).

The major contributions of this paper are as follows.

- We develop a self-supervised fine-tuning algorithm to make existing ASR systems robust. This method only requires fine-tuning the encoder of the STT model. Furthermore, our approach does not require any human annotations (i.e. transcriptions) for training.
- We present a new signal processing attack and evaluate our defence against the new attack and other state-of-the-art attacks [5, 6]. The new attack was developed to test our defence in more attack scenarios in addition to the two existing attacks [5, 6].
- We develop a signal processing attack detection model to handle benign audio samples so that the benign samples are not put through the defence mechanism.

- We experimentally evaluate the performance of the end-to-end defence, which is the combination of fine-tuning algorithm and the detection model.

The rest of the paper is organized as follows. Section 2 surveys previous related efforts. Section 3 introduces a new signal processing attack. Section 4 describes our self-supervised fine-tuning algorithm and the architecture of the attack detection network. Sections 5 and 6 present the experimental setup and results, respectively, followed by the conclusion in Section 7. The appendix of the paper includes survey results (Appendix A.1) and experiments in noisy environments (Appendix A.2).

The code used to train and evaluate Robust Whisper and benign sample detector is available at GitHub. The datasets are available at Hugging Face.

2 BACKGROUND AND RELATED WORK

2.1 ASR Systems

ASR systems comprise of three main steps: preprocessing, feature extraction, and decoding. Preprocessing engages in removing the background noises, interference, and other disturbing components of the audio file. Feature Extraction retains only the important information using various signal processing techniques such as Discrete Fourier Transforms (DFT), Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding, and the Perceptual Linear Prediction method. In addition to that, machine learning extraction layers are trained to learn which features are to be extracted. During the decoding phase, the extracted features are fed to the decoding model and this returns the corresponding transcription. Models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Hidden Markov Models, and Gaussian Mixture Models have been used in ASR systems. Radford et al. [26] introduced a method to use encoder-decoder transformer architecture introduced by Vaswani et al. [31] to come up with a robust ASR system. This model is trained on a large corpus of speech-text data in a multi-task setting. They use weak supervision which incorporates a text standardisation step before calculating the loss function during training. On a high-level interpretation, the transformer encoder compresses the audio waveform into a latent space. The decoder then uses this encoding to generate the text auto-regressively.

2.2 Adversarial Attacks in Audio Domain

2.2.1 Optimization-based attacks. Recent research has proven that ASR systems are vulnerable to adversarial attacks. L_p clipping has been widely used in magnitude controlling of attack perturbations and has shown to be successful in the image domain [21, 22]. However, it is a poor technique for controlling the imperceptibility of a perturbation in the audio domain as it introduces undesirable audio effects. Going beyond L_p clipping method, researchers have identified various methods to generate adversarial examples. Carlini et al. proposed a white-box iterative optimization-based attack that could produce targeted adversarial audio waveforms [13]. Given the original waveform x, they produced $x + \delta$, which is 99.9% similar to the original (x) but transcribed to any phrase of choice. The perturbation δ is nearly inaudible. In [25], Qin et al. construct imperceptible audio adversarial examples using the psychoacoustic principle of auditory masking. Their attack occurs in two optimization stages. In the first stage, they focus on finding a comparatively small perturbation that can mislead the network. This was done following the method presented in [13]. The second stage focuses on making the adversarial example imperceptible. In [7], Abdullah et al. proposed an equalization-based psycho-acoustic attack that can fool traditional as well as fully end-to-end ASRs, unlike the existing psycho-acoustic attacks, which could only be applied against traditional models. Moreover, their work showed evidence that their attack is less noisy than the L_p clipping method.

2.2.2 Gradient-free attacks. Alzantot et al. presented a black box targeted attack crafted using an approach based on gradient-free genetic algorithms [10]. The algorithm creates a population of potential adversarial examples by adding random noise to a subset of samples within the given audio clip. Each generated example in the population is assigned a fitness score based on how well it fools the target model into predicting the desired transcription. The next generation of adversarial examples is produced through a process of selecting examples with higher fitness scores and crossover, which involves combining pairs of population members to create new "children" examples for potential improvements and mutation. Mutation introduces occasional tiny random noises to the children, further diversifying the population. This iterative process continues for a predetermined number of epochs or until the attack successfully deceives the target model. This genetic algorithm-based method does not need any knowledge of the target model architecture or parameters. Taori et al. proposed a black box adversarial perturbation method that combines the approaches of both genetic algorithm and gradient estimation [30]. The initial phase of the attack employs genetic algorithms to generate a suitable sample. To mitigate excessive mutations and noise, a novel momentum mutation update is integrated into the standard genetic algorithm. In the subsequent phase, gradient estimation is used. This involves estimating gradients for individual audio points and enhancing the precision of noise insertion as the adversarial example approaches its target.

2.2.3 Signal Processing attacks. These attacks are unique to the audio domain. Abdullah et al. proposed an efficient and transferable black-box attack named Kenansville attack that can fool any state-of-the-art speech recognition and voice identification system in near real-time with fewer queries [6]. The attack does not degrade the quality of the audio and the introduced changes are imperceptible to humans. They identified that their attack is robust to existing adversarial attack detection and defence mechanisms. After evaluating many latest attacks on ASRs, Abdulla et al. identified the Kenansville attack as the best attack to generate CAPTCHAs due to its high transferability [5]. They also identified that adding white noise defends a STT model against the Kenansville attack. As a solution, the same authors proposed the Yeehaw Junction attack, an improved version of the Kenansville attack to design robust audio CAPTCHAs [5]. There they added some extra features in addition to the decimation done in the initial attack. The additions are, adding Gaussian noise to the perturbed sample and clipping the large amplitudes of dominant frequencies preserving the location of the frequency peaks so that the audio remains highly intelligible to the human ear. In the next section (Section 2.3), we go into detail about the underlying mechanisms of the Kenansville and

Yeehaw Junction attacks since our defence is tested against these two state-of-the-art attacks.

2.3 Details of Two State-of-the-Art Attacks

2.3.1 Kenansville attack. Abdullah et al. have noted that existing attacks on ASR systems fall short of being truly effective [6]. To address this, they propose a novel attack named "Kenansville", which operates by augmenting frequency components in the signal following its frequency domain representation. However, this method shows limited efficacy against temporal dependency-based techniques employed for adversarial detection and defence [36]. Their approach is primarily based on the hypothesis that ASR and Automatic Voice Identification systems rely on speech components that are non-essential for human comprehension. The process involves taking the DFT of the audio signal, subsequently eliminating frequency components with intensities below a predetermined threshold value from the spectrum. By applying the Inverse Fourier Transformation, the original time-domain audio signal is reconstructed.

The most crucial aspect lies in selecting an appropriate threshold value, as maintaining imperceptibility after removing frequency components is paramount. If the threshold value is set too high, the audio quality degrades significantly, making it difficult for both the model and human listeners to interpret the reconstructed audio correctly. Conversely, if the threshold value is too low, both human listeners and the STT model can readily comprehend the reconstructed audio. To identify the optimal threshold value, a binary search is conducted between the maximum and minimum intensities of the DFT-transformed signal. During the execution of the attack, if the model output matches the original transcription, the method increases the threshold value and feeds it back to the model. Conversely, if the transcriptions differ, the method reduces the threshold value and provides the updated value to the model. It is important to mention the exit condition of the binary search algorithm. As we mentioned earlier, one possible exit condition is checking if the transcription differs. Change in a single word will stop the binary search, but if you need a much stronger attack, you can set the exit condition to check until a certain number of words change. At the same time, you should be mindful of the amount of distortion it adds since there is always a trade-off.

2.3.2 Yeehaw Junction attack. The Yeehaw Junction attack represents a significant extension of the Kenansville attack [5]. Abdullah et al. discovered that by adding power to the empty frequency bins, which were removed in the priorly proposed Kenansville process [6], they could effectively counteract the effects of the attack. This addition of white gaussian noise increases the power evenly across all frequency bins, rendering the audio CAPTCHAs intelligible to humans while forcing STT models to output empty transcriptions. To evaluate the success of Kenansville attack against optimization attacks, the authors employed the Levenshtein distance score between the phonetic representation of the original and attacked audio samples as a metric for phonetic similarity. Notably, the Kenansville attack produced the highest distance between the original and attacked audio samples.

Since the Kenansville attack failed to generate intelligible audio CAPTCHAs against an adaptive adversary, Abdullah et al. devised the Yeehaw Junction attack, which involves a process of decimation, clipping, and noising [5]. The decimation step follows the same process as discussed in the Kenansville approach. The "spectral clipping" method exploits the fact that the human ear relies on specific dominant frequency bands, known as formants, to identify individual phonemes. By clipping these dominant frequencies in the spectrogram, the method creates phonetic structures that do not occur naturally. This clever technique effectively tricks STT models while remaining imperceptible to human listeners, as the location of the frequency bands in the spectrum remains unchanged. Furthermore, the clipping approach proves to be robust against Gaussian noise-based adaptive adversaries. Clipping evens out the peak structure, which is crucial for the ASR system to transcribe accurately. The addition of random noise fails to recreate the clipped-out peak structure, preventing the adversary from obtaining the correct original transcript.

Similar to the Kenansville attack, the Yeehaw Junction attack also employs the binary search algorithm to select the optimal clipping threshold. During the execution of the attack, if the model output matches the original transcription, the method decreases the clipping threshold value and feeds it back to the model. Conversely, if the transcriptions differ, the method increases the threshold value and provides the updated value to the model. In response to the adaptive adversary during audio CAPTCHA generation, the defence pipeline involves adding noise to every audio sample before passing it to the STT model.

2.4 Transferability of Adversarial Attacks in ASR Systems

Transferability of adversarial attacks enables attackers to deploy attacks on ASR systems under a black-box setting which is the most practical scenario in the real world. Abdullah et al. experimentally demonstrated that transferability of optimization attacks against STT models is highly unlikely even under situations where both shadow and target models share the same architecture, hyperparameters, random seed and training data [8]. Input type, MFCC, RNN, output type, vocabulary and sequence size were identified as the factors that affect the targeted transferability of optimization attacks [4]. As optimization attacks do not provide targeted transferability, the community began to focus on signal processing attacks which provide targeted transferability. But still, clean, targeted signal processing attacks do not exist. Unlike ASR systems, speaker recognition systems are not robust to transferability. Abdullah et al. proposed using an ASR for text verification in the speaker recognition pipeline as a measure to ensure the robustness of the overall speaker recognition pipeline [4].

2.5 Defences against Adversarial Attacks on ASR Systems

In this section, we discuss the existing defences under two main categories: i) detection and ii) filtering.

2.5.1 Detection. Hussain et al. proposed a framework called WaveGuard to detect adversarial inputs from benign inputs using audio transformation functions (e.g. down-sampling and up-sampling,

quantization and dequantization, filtering, Mel spectrogram extraction and inversion) and by analyzing the ASR transcriptions of the original and transformed audio [19]. The proposed framework demonstrated reliable detection of adversarial examples and robustness even towards adaptive adversaries who have complete knowledge of their defence. In [36], Yang et al. proposed an adversarial audio detection mechanism based on temporal dependencies. Given an input audio sample, the audio sample is partitioned into two. Then the first partition and the entire audio sample are fed to the STT model to get the transcriptions. If the corresponding parts of the transcriptions are similar, the audio is detected as benign. If not, the audio is adversarial. The detection is done based on the premise that adversarial attacks distort the temporal dependence within the audio. The authors claim that the temporal dependency-based approach is lightweight, simple and highly effective at detecting traditional adversarial attacks [36].

The above methods need pre-processing of the audio sample and need to consider the transcription of the audio sample to arrive at the decision. This is a time-consuming process and is the main limitation of the existing classical detection methods.

2.5.2 Filtering. Defence mechanisms such as adversarial training and convex relaxations [33] are harder to be used in speech recognition. Olivier et al. proposed a defence based on randomized smoothing for speech recognition systems which is robust to all the attacks that use inaudible noise [23]. Eisenhofer et al. proposed a method to tame audio adversarial attacks on ASRs by applying psychoacoustic principles [17]. They proposed to modify the existing ASR systems by (i) adding psychoacoustic filtering to remove the inaudible parts of the input audio and (ii) applying a band-pass filter after the feature extraction layer to remove the lower and higher frequencies of the audio signal and training the STT models with augmented data. Through this mechanism, they showed that ASR systems learn a better approximation of human perception and adversaries are forced to bring any adversarial perturbation into audible ranges.

At present, the most prevalent defence mechanism in audio processing is denoising which removes/reduces perturbations in audio. One commonly used technique for denoising is autoencoders. Wu et al. proposed Mockingjay, which utilizes bidirectional transformer encoders [29]. The model learns to reconstruct or predict the original frames given masked frames during training. In [28], Wu et al. proposed Transformer Encoder Representations from Alteration (TERA), a more advanced self-supervised model compared to Mockingjay which utilises alteration along three orthogonal axes (time, frequency and magnitude) to pre-train transformer encoders. Sreeram et al. propose a denoiser based on the DEMUCS architecture that is independent of the downstream ASR pipeline [11]. They found that training the denoiser with a perceptually motivated loss increases the adversarial robustness without affecting the benign audio samples. The authors adopted the pre-trained DEMUCS-based denoising model, presented by Defossez et al. in [9]. DEMUCS architecture is an encoder-decoder-based deep neural network with U-Net skip connections and a sequence modelling network which is developed for music-source separation in the waveform domain [16]. Defossez et al. have shown that it could successfully be converted into a casual speech enhancer, processing speech waveforms in real-time on consumer-level CPU [9].

3 A NEW SIGNAL PROCESSING ATTACK

A variety of attacks is required to evaluate the robustness of our defence mechanism. However, previous work includes only two state-of-the-art signal processing attacks. Therefore, we propose a new signal processing attack by combining frequency decimation and imaginary component clipping methods. The decimation steps are the same as what is outlined in the Yeehaw Junction attack [5]. Through several experiments, we identified that the optimal decimation threshold exhibits a parabolic pattern. When the decimation threshold is set too high, substantial noise arises due to decimation before entering the clipping process, resulting in identifiable perturbations to the audio. On the other hand, if the decimation threshold is set too low, there are no significant frequency bins removed. We also observed that the clipping effect greatly influences the audio quality. Hence, the new attack focuses on determining an optimized decimation threshold based on experimental values following a parabolic pattern.

The decision to clip only the imaginary components of the audio sample stems from the understanding that the imaginary component of a complex value captures the contribution of sine waves of different frequencies to the audio. In a manner similar to the Yeehaw Junction clipping process, we selectively clip the imaginary components after obtaining the DFT of the audio, which provides both the real and imaginary components.

Clipping the imaginary components alters the Power Spectral Density of the audio, resulting in indirect changes to the audio phase. By focusing on the imaginary components, we manipulate the audio in a way that effectively deceives ASR systems while maintaining imperceptibility to human listeners.

Suppose the DFT of the audio sample is denoted as $x + jy$, where j denotes the imaginary component. Then clipping only the imaginary component changes the DFT as follows:

$$x + jy \rightarrow x + jy\frac{k_1}{k_2} \tag{1}$$

k_1 and k_2 are constants. k_1 is the absolute threshold value for clipping and k_2 is the absolute value of the imaginary component ($k_2 = |y|$). Imaginary component clipping changes the phase of each frequency component as shown in the equation below:

$$\phi = \tan(\frac{y}{x}) \ \rightarrow \phi = \tan(\frac{y \cdot k_1}{k_2 \cdot x}) \tag{2}$$

Hence, the main stages in crafting the new attack can be identified as (i) decomposing the original audio waveform to its frequency components, (ii) decimating the low-intensity frequencies, (iii) separating the real and imaginary components of frequency components (iv) clipping only the imaginary component based on a threshold value, (v) reconstructing a raw audio waveform concatenating real imaginary components, (iv) evaluating it by sending through an ASR System. This process is repetitively done, alternating the clipping threshold at each iteration until the attack is successful. Figure 3 provides a detailed overview of the process of creating the new attack. Figure 4 elaborates its architecture, which is the separation of real and imaginary components of the DFT decomposition and clipping only the imaginary part using a threshold before constructing the final waveform. These visual representations help demonstrate the intricate steps involved in our approach. Since we

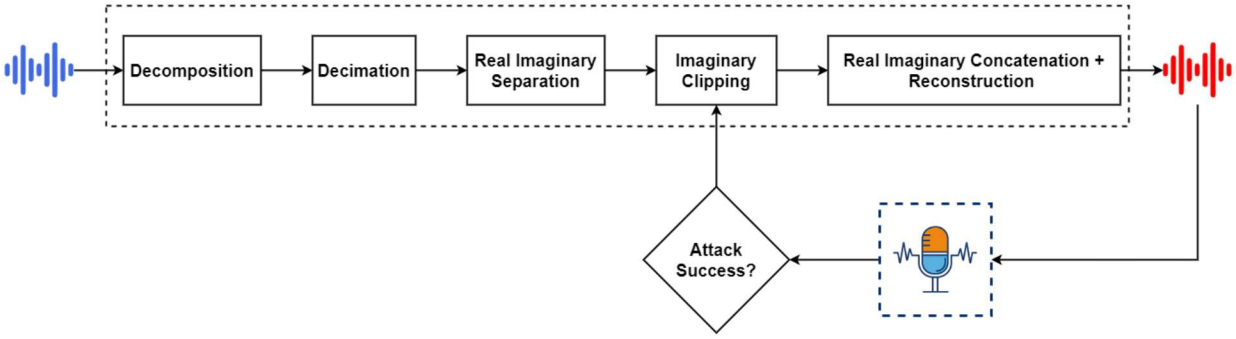

Figure 3: Overview of the new attack. First, the audio sample is decomposed into its frequency components and decimated by discarding low-intensity frequencies. Then, the real and imaginary components are separated. Next, the imaginary component is clipped based on a threshold. Afterwards, the real and imaginary components are concatenated back and reconstructed the modified spectrum into a raw audio sample. Then, pass it to the ASR and check for success. If it is not successful, then lower the clipping threshold and re-run.

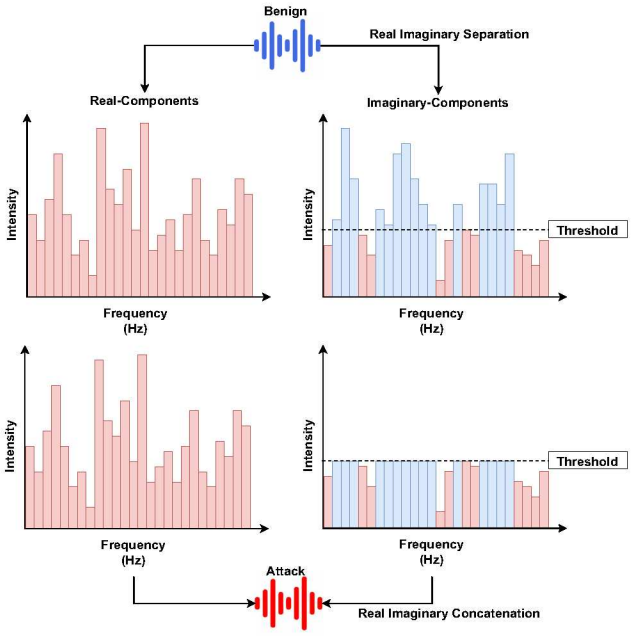

Figure 4: Architecture of new attack. The DFT decomposition gives both the real and imaginary components. We separate them and clip only the imaginary component using the optimal threshold. To find the optimal threshold, a similar approach as in the Yeehaw Junction attack is followed. Finally, the real and imaginary components are concatenated back to construct the final waveform.

are clipping only the imaginary component, hereinafter, the new attack will be referred to as the "Imaginary Clipping Attack".

We investigate the imperceptibility of the Imaginary Clipping Attack and compare it with state-of-the-art attacks through a survey. Additional information about the survey and the analysis of its results are presented in Appendix A.1.

4 METHODOLOGY

4.1 Whisper Architecture

In [26], Radford et al. introduced a method to use encoder-decoder transformer architecture introduced by Vaswani et al. to come up with a robust STT model [31]. This model is trained on a large corpus of speech-text data in a multi-task setting. They use weak supervision, which incorporates a text standardization step before calculating the loss function during training. On a high-level interpretation, the transformer encoder compresses the audio waveform into a latent space. The decoder uses this encoding to auto-regressively generate the text. We hypothesize that errors happening in the encoder propagate into the decoder, resulting in errors in the transcription. This is illustrated in Figure 5. In the figure, we have plotted original-attacked pairs in the latent space. We can observe that transcription errors are represented in the latent space. We hypothesize that by making the encoder robust, we can reduce errors in the transcription. This saves the cost of fine-tuning the entire end-to-end ASR pipeline to make it robust to new attacks.

4.2 Our Approach - Robust Whisper

Inspired by the work of Zhao et al., we explore the idea of using a latent space [37]. Training two additional neural networks is required to incorporate this idea — (i) an encoder that can map the continuous waveform to a latent vector and (ii) a decoder that can map a latent vector to the discrete space (text). Once we train this encoder-decoder network, we can get rid of the decoder and use the encoder output as the representation of both continuous and discrete spaces. In order to train the audio-to-audio reconstruction network, we can create a pipeline consisting of the audio-to-audio reconstruction network followed by an audio-to-latent encoder we trained

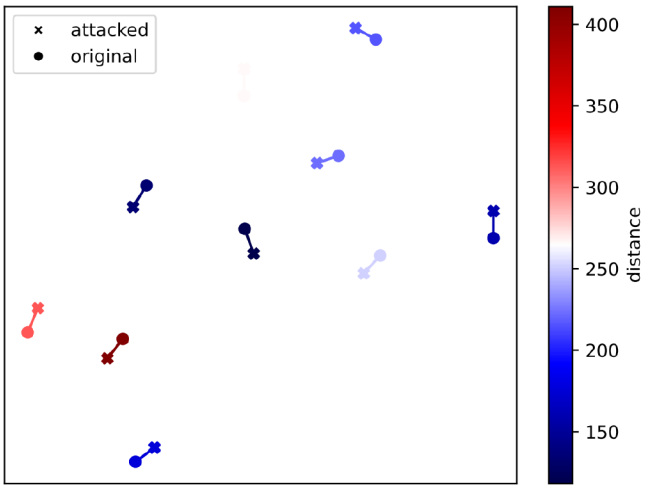

Figure 5: Whisper encoder latent space. The figure shows nine pairs of attacked-benign audios in the Whisper encoder latent space. We projected the high-dimensional latent space into a two-dimensional space using the t-SNE algorithm. We color-mapped the L_2 distances between each pair. The distances illustrated in the figure do not align with the actual distance due to the t-SNE algorithm.

previously. Now, the objective function is transformed to be "minimizing the distance between the latent representation of the original waveform and the adversarial waveform". Since this latent representation contains information about the text, the objective function will reduce the WER. However, this method requires training an encoder-decoder network in addition to the reconstruction network. But the latest ASR systems based on encoder-decoder transformer models already have an information-rich latent space [26]. The output embedding of the transformer encoder carries enough information such that the decoder can auto-regressively generate the text using it. Thus, we can use the encoder output space as the latent space.

Given this latent space, there are two possible approaches to building a defence. The first approach is to use this latent space to train a reconstruction network as described earlier. This approach has a drawback. Adding an additional reconstruction network to the ASR pipeline adds extra latency, disabling or hampering real-time applications. The second approach is making the latent space robust such that an adversarial waveform has the same latent representation as its original counterpart. This does not add any additional latency to the pipeline, and therefore, we explore the latter as our solution.

According to our hypothesis in Section 4.1, our approach only requires fine-tuning the transformer encoder. The fine-tuning approach is illustrated in Figure 6. For each sample x we generate an adversarial counterpart x'. Then we apply log mel spectrogram transformation to both x and x' and the resulting outputs are named as x_{mel} and x'_{mel} respectively.

In order to fine-tune the encoder, we need two copies of the existing encoder from the STT model we are going to apply the defence. From the two copies, we freeze the weights of one (f_θ) and keep

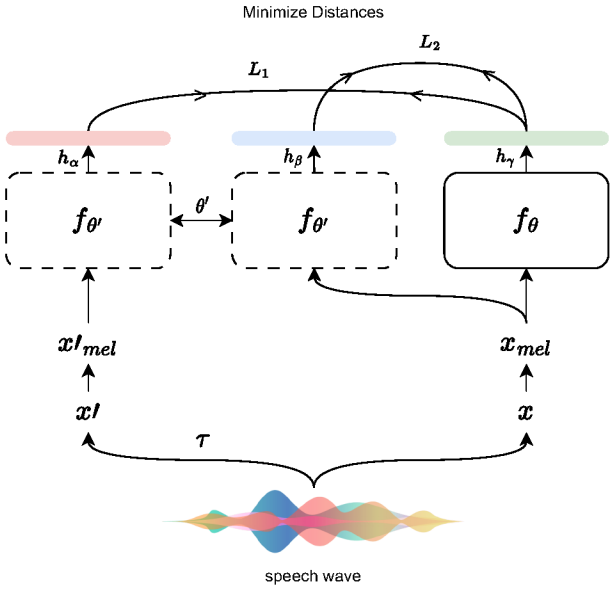

Figure 6: Fine-tuning approach.

the other as trainable ($f_{\theta'}$). Then we generate three embeddings as follows;

$$\begin{aligned} h_\alpha &= f_{\theta'}(x'_{mel}) \\ h_\beta &= f_{\theta'}(x_{mel}) \\ h_\gamma &= f_\theta(x_{mel}) \end{aligned} \tag{3}$$

We need to bring the latent vector of an attacked speech audio generated by the fine-tuned encoder (i.e. h_α) closer to its counterparts' (i.e. benign audio) latent vector generated by the original encoder (h_γ). We define our first loss for this task.

$$L_1 = \|h_\alpha - h_\gamma\|_{L_2} \tag{4}$$

But this loss alone cannot achieve the expected behaviour. This will make the encoder robust to attacked audios. But there is no guarantee about what will happen to the latent vectors of benign audios. We need to make the latent vector of the benign sample stationary. We use the second loss to achieve this.

$$L_2 = \|h_\beta - h_\gamma\|_{L_2} \tag{5}$$

We use the sum of two losses to upgrade the weights of the trainable encoder ($f_{\theta'}$). The fine-tuning algorithm is given in the algorithm 1.

In the training process, we keep a copy of the original encoder as a reference (f_θ). As a result, the minimum WER that can be achieved by the fined-tuned encoder ($f_{\theta'}$) on adversarial samples (D') is greater than or equal to the WER achieved by the original encoder (f_θ) on benign samples (D);

$$\text{WER}(\theta', D') \geq \text{WER}(\theta, D) \tag{6}$$

Where $\text{WER}(\theta, D)$ is the upper bound for the performance.

We determined the number of layers to fine-tune experimentally. We achieved the best results by only fine-tuning the two convolutional layers of the Whisper [26] encoder. During the fine-tuning, we froze all the multi-headed attention layers of the encoder and only kept the convolutional layers as trainable.

Algorithm 1: Encoder fine-tuning

Requires: An encoder f_θ from an STT model, adversarial attack $\tau\cdot$, speech samples dataset D.
Hyper-parameters: Number of epochs N. Gradient step size (learning rate) λ
$f_{\theta'} \leftarrow f_\theta$;
for $i = 0, 1, 2, ..., N$ **do**
  for $x \in D$ **do**
    $x' \leftarrow \tau(x)$;
    $x_{mel} \leftarrow mel(x)$;
    $x'_{mel} \leftarrow mel(x')$;
    $h_\alpha \leftarrow f'_\theta(x'_{mel})$;
    $h_\beta \leftarrow f'_\theta(x_{mel})$;
    $h_\gamma \leftarrow f_\theta(x_{mel})$;
    $L_1 = \|h_\alpha - h_\gamma\|_{L_2}$;
    $L_2 = \|h_\beta - h_\gamma\|_{L_2}$;
    $L = L_1 + L_2$;
    $\theta' \leftarrow \theta' - \lambda\nabla_{\theta'}L$
  end
end

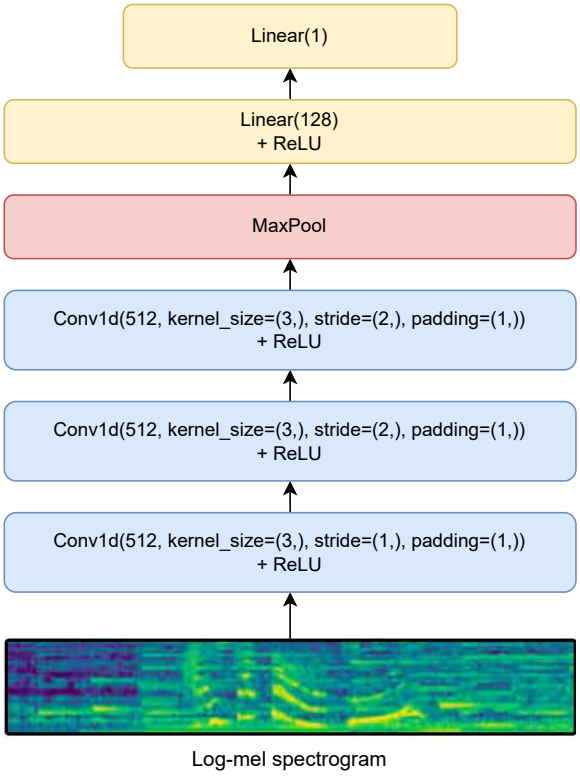

Figure 7: Detector architecture.

4.3 Benign Sample Detection

Even though we designed our self-supervised fine-tuning loss to reduce the damages that happen to transcriptions of benign samples, there could still be small errors. We must ensure that the proposed Robust Whisper approach would not cause errors with the benign samples. Ideally, we expect only the attacked samples to be reconstructed using Robust Whisper. To achieve this, we developed a benign sample detector to detect whether a given audio sample is attacked or benign. Only the samples that are classified as "attacked" are fed into Robust Whisper. The benign samples are fed into the default Vanilla Whisper model, which is not fine-tuned with our proposed defence mechanism.

The detector is a binary classifier. The architecture is given in Figure 7. First, the log-mel spectrogram of the audio is taken. For that, the Short-Time Fourier Transform (STFT) of the audio signal is generated, and then a mel filterbank is applied to the STFT magnitudes. Here, the mel scale is non-linear and approximates how the pitch variations are perceived by humans. After that, the magnitudes are transformed into a logarithmic scale. The reason for this is that the way that people perceive loudness is logarithmic.

We considered the log-mel spectrogram because it offers a means of representing the frequency content of an audio signal in a manner that is similar to how people perceive sound. This makes it simpler to process and analyse audio data. Then the generated log-mel spectrogram is sent through three 1-D convolution layers with Rectified Linear Unit (ReLU) activation. Convolution layers are selected to detect temporal dependencies of the input. Next, the output is sent through a max pool layer, followed by a linear layer with ReLU activation, and finally through a linear layer again. We considered small kernel sizes and shallow architecture to prevent the model from overfitting to underlying words and to learn more global features of the audio. We also randomly added Additive White Gaussian Noise (AWGN) to benign samples to avoid overfitting and make the model more robust. We did not add AWGN to attacked samples since they already have enough distortions and Yeehaw Junction attack adds AWGN. We introduced randomness by randomly selecting the SNR from a uniform distribution. We also randomly left half of the benign samples without adding any noise. The target of learning as much as global features is the reason for the higher number of filters of the convolutional layers.

5 EXPERIMENTAL SETUP

5.1 Datasets

For training and testing, we created our own signal processing attack datasets. To create these datasets we used two publicly available datasets used in the literature for training and testing STT models. Panayotov et al. [24] created the Librishpeech dataset using a large corpus of public-domain audiobooks that are part of the LibriVox project. We used non-overlapping portions of the train-clean-100 subset of Librispeech for our datasets. Ardila et al. [12] created the Common Voice dataset, which is a large multilingual

speech corpus by crowd-sourcing. We used multiple subsets (multiple versions of Delta Segments) of the Common Voice dataset to create our datasets. We randomly picked samples for these subsets in a way there are no duplicates and no overlaps between subsets. We ran the three attack algorithms Kenansville, Yeehaw Junction, and Imaginary clipping, on the previously mentioned datasets to create the adversarial audios. As explained in the sections 2.3 and 3, the attack algorithms require an ASR system to query. We use Whisper [26] and AssemblyAI [1] for this purpose. A summary of the datasets is given in Table 1. The dataset names have the following format

$$\{task\}\_\{attacked\_model\}\_\{source\_dataset\}$$

Table 1: Dataset summary.

Dataset Name	Number of Samples
cl_whisper_librispeech	18000
cl_assembly_librispeech	300
cl_whisper_commonvoice	300
cl_assembly_commonvoice	300
ae_whisper_librispeech	9000
ae_assembly_librispeech	150
ae_whisper_commonvoice	150
ae_assembly_commonvoice	150

cl (classification) and *ae* (auto-encorder) denote the tasks. *cl* datasets are used for training and testing the detector, while *ae* datasets are used for training and testing the encoder. *attacked_model* refers to the STT model that was queried to create the adversarial samples.

For the training, validation, and testing of the encoder, we use 60%, 20%, and 20% of the *ae_whisper_librispeech*, respectively. Additionally, we use *ae_assembly_librispeech*, *ae_whisper_commonvoice*, and *ae_assembly_commonvoice* for testing the encoder. Similarly, for the training, validation, and testing of the detector, we use 60%, 20%, and 20% of the *cl_whisper_librispeech*, respectively. Additionally, we use *ae_assembly_librispeech*, *ae_whisper_commonvoice*, and *ae_assembly_commonvoice* for testing the detector.

5.2 Experiments

Our experiments can be mainly categorized into two sections, experiments conducted to compare between the attacks and experiments conducted to evaluate the defence mechanism.

5.2.1 Experiments to compare between attacks. We initially conducted a separate experiment to compare the attacks; Kenansville, Yeehaw Junction, and the new attack we developed, which is the Imaginary Clipping attack. The dataset we used was the publicly available common voice dataset, and we selected 50 random data samples from the dataset. Data samples were sent through an attack algorithm querying an STT model until it generates the best possible perturbed audio which is the least perceptible to the human ear. For each data sample, we obtained results using the three attack methods and three ASR query engines. The attack procedure is as mentioned in Section 3. For the query purpose, the three ASRs — AssemblyAI, OpenAI Whisper, and Google Cloud Speech API were utilized. The results of the experiments are described in Section 6.1.

5.2.2 Experiments to evaluate the defence mechanism. We conduct experiments for the encoder, detector, and end-to-end defence under four scenarios. These four scenarios are designed in increasing order of difficulty of the transferability for the defence. All the adversarial data used for training was generated by querying Whisper [26] STT model using audio files from the source dataset *librispeech* (i.e. *{task}_whisper_librispeech*). In each scenario, we change either *attacked_model*, *source_dataset*, or both.

- **Scenario 1**: This is the least challenging scenario for our defence. In this scenario, we evaluate our models on test data coming from the same distribution as training data. We use 20% of *{task}_whisper_librispeech* dataset.
- **Scenario 2**: In this scenario we only change the *attacked_model* to AssemblyAI [1] (i.e *{task}_assembly_librispeech*). This is comparatively a difficult task compared to Scenario 1 as the targeted model is different. At the same time, we observed AssemblyAI is a much more robust model than Whisper. This causes attacks to add higher perturbations to audio, making it harder to be reconstructed.
- **Scenario 3**: In this scenario we only change the *source_dataset* to Common Voice while keeping the *attacked_model* similar to that of training data (i.e *{task}_whisper_commonvoice*). This task is comparatively more difficult than Scenario 1 as the audio distribution is drastically different from what the model is trained on.
- **Scenario 4**: In this scenario we change both *attacked_model* and *source_dataset* resulting in the most difficult scenario.(i.e *{task}_assembly_commonvoice*). Since this is the combination of Scenarios 2 and 3, it is more difficult than all previous scenarios.

The *{task}* parameter of each scenario changes accordingly whether we are evaluating the encoder, detector, or the end-to-end defence. The results of the experiments are analyzed in Section 6.

6 RESULTS

In this section, we first outline a comparison between different attacks including the new attack - Imaginary Clipping (Section 6.1) and then discuss results of Robust Whisper (Section 6.2).

6.1 Comparison Between Attacks

As described in Section 5.2.1, the results obtained by comparing attacks using three different ASRs are tabulated in Table 2. The metrics we used for the comparison are Mean Squared Error (MSE) and the WER. The MSE is the L2 distance between the frequency components of the original and the perturbed audio. A lower MSE indicates less imperceptibility of the perturbation to the human ear. WER represents the percentage of words that the ASR incorrectly predicted for the perturbed audio referenced to the original transcription. For the computation of WER, we used the pytorch *WordErrorRate* Module. For a successful attack, the WER should be higher. We recorded the median value for the MSE and the WER to avoid the effect of outliers. Additionally, we recorded the success rate of each attack. The success rate indicates how many samples

out of the 50 were actually attacked. We did this analysis because some attacks may end up not changing the transcriptions at all, resulting in an unsuccessful attack. The cause for this is the upper bound we set for the distortion the attack algorithm is allowed to add while querying the model.

Table 2: Attack success rate with 50 audio samples. Median MSE and MSR are also outlined.

Attacks	AssemblyAI			Whisper			Google Cloud		
	Success Rate	WER	MSE	Success Rate	WER	MSE	Success Rate	WER	MSE
Kenansville	**50/50**	0.106	**18.039**	**50/50**	0.134	**0.646**	46/50	0.183	**0.001**
Yeehaw Junction	39/50	**0.111**	56.147	49/50	0.4	9.215	**50/50**	**0.833**	8.863
Imaginary Clipping	32/50	0.101	38.85	47/50	**0.417**	10.4	49/50	**0.833**	8.976

Kenansville shows the highest success rate across all three ASRs. It shows the least MSE, but also it has the least WER as well. Yeehaw Junction attack has a high MSE. The new attack also has a high MSE but comparatively, it is lesser than the Yeehaw. Observing the WERs of the Yeehaw attack and the new attack, it can be stated that the new attack shows competitive results to the Yeehaw attack with a relatively high WER.

Although the purpose for developing a new attack was to evaluate the robustness of our defence mechanism, the Imaginary Clipping attack is competitive compared to existing attacks. So it may be of interest to explore the attack as future work beyond the current paper.

6.2 Robust Whisper

Robust Whisper is compared with three baselines. (i) With the undefended Whisper model. (ii) In [11], Sreeram et al. introduced DEMUCS-Denoiser, a fine-tuned version of DEMUCS speech enhancement model [9], against adversarial attacks. The fine-tuning was done using a perceptual loss. When evaluating this method, we create a pipeline. First, the audio is sent through the filtering model (DEMUCS-denoiser) and then the filtered audio is sent to the undefended Whisper model. (iii) MetricGan+ [18] is the most downloaded speech enhancement model available on Hugging Face. MetricGan+ is not fine-tuned against adversarial attacks. We use it as a baseline for speech enhancement models to show that an off-the-shelf speech enhancement model is not able to filter out the adversarial perturbations. We use the same pipeline used for the DEMUCS-denoiser to evaluate MetricGan+. Table 3 summarizes the results according to the four scenarios explained in Section 5.2. In three out of the four scenarios, our approach gave the best results. DEMUCS-denoiser, which is fine-tuned using a perceptual task, underperforms our method in three scenarios. This supports the hypothesis of how the disjoint nature of tasks while training can affect performance. The results also show how even a state-of-the-art speech enhancement model not only underperforms but also makes the transcriptions much worse. Qualitative results are shown in Table 4.

It is important to note that we consider the transcription of the benign audio taken from the original Whisper ASR as the ground truth. This is applicable to both quantitative and qualitative results in the paper. All the WERs are calculated based on this. The main reason for this is the objective of the defence. The objective of the defence is to make the model robust, in other words reducing the variance of the output against perturbations added by the attacks. That is why the lower bound of the WER is WER_θ as explained in Section 4.2.

Table 3: Results of text reconstruction. Robust Whisper is compared with the undefended Whisper model, a state-of-the-art speech enhancement model, and an audio source separation model fine-tuned with a perceptual loss (DEMUCS-denoiser) on the task of text reconstruction. We evaluated results under four scenarios explained in Section 5.2.2. The recorded WERs are averaged values across each test dataset.

	Dataset	Whisper_ LibriSpeech	Whisper_ CommonVoice	AssemblyAI_ Librispeech	AssemblyAI_ CommonVoice
Attacked	Undefended	0.10419	0.4370	0.3535	0.8281
	speechbrain-MetricGan+	0.1539	0.5471	0.4105	1.1617
	DEMUCS-denoiser	0.1099	0.4127	**0.2901**	1.4431
	RobustWhisper	**0.0730**	**0.3630**	0.3002	**0.5937**
Benign	speechbrain-MetricGan+	0.0573	0.1957	0.0516	0.2460
	DEMUCS-denoiser	0.0252	0.1584	**0.0214**	0.0995
	RobustWhisper	**0.0167**	**0.1453**	0.0273	**0.0783**

Another important point to note is the WER for benign samples in our method. Even though we used the second loss L_2 given in Equation 5, we can still observe a small WER for the benign samples when using Robust Whisper. This can be reduced by using more training data. However, our focus is to find a method to adapt STT models to new attacks quickly and with a minimum amount of data. So, as we discussed in the section 4.3, we build an attack detection model to handle benign audio samples so that the benign samples are not put through Robust Whisper. The results of the accuracy of the detector is analyzed in the next section.

6.3 Benign Sample Detection

Since signal processing attacks are a new domain, and due to the limitations of publicly available datasets, there are no fixed machine learning-based detectors for signal processing attacks. So, we checked the results for non-parametric classical methods, which were discussed in Section 2.5.1. This includes temporal dependency [36] and three types of audio transformations named down-sampling and up-sampling, quantization and dequantization, and filtering [19]. We compared our method with these existing methods under the four scenarios discussed in Section 5.2.2. We used the AUC score and average time per sample as the comparing metrics. The results are shown in Table 5. Our detector gives the best performance for all four scenarios.

Further, for our classifier, the accuracy, precision, recall, and F1 score metrics were calculated for the four different test scenarios in Section 5.2.2. Table 6 shows the calculated results. As we explained in the section 5.2.2, the increasing difficulty has affected the performance of the detector. The detector has the best performance in the first scenario and the worst performance in the fourth scenario, which are the most difficult and least difficult scenarios, respectively. The results show that the detector finds scenario 3 more difficult than scenario 2. This is mainly caused by the high distortions added to the audio by the attacks due to the high robustness of the AssemblyAI STT model. The high distortion makes it easier for the detector to detect the attack. Overall, the results prove that

Table 4: Comparison of Transcriptions. In the first column, we show the transcriptions of benign audio taken from the undefended Whisper. In the second column, we show the changes that happen to the transcriptions when the attacked version of audio is given to the undefended Whisper model. In the last column, we show the transcriptions of the attacked audios taken from the Robust Whisper.

Original Transcription	Transcription from undefended Whisper	Transcription from Robust Whisper
and also try drawing for engraving.	I mean, also try drawing for a reading	Men also try drawing for engraving.
She was educated in Alfred University.	She loves and gives me the ultimate beauty of life.	She was in Britain in Alfred, University.
But he never lost his love of motorcycle racing.	But he never lost his love of notice, I thought racing	But he never lost his love of motorcycling racing.
He was married but had separated from his wife.	He was married but had subraised him from his wife.	He was married but had separated from his wife.

Table 5: Detection results. The performance of our detector is compared with four existing classical detector methods under four scenarios in terms of AUC score and average time per sample.

Detector Method	Whisper_LibriSpeech		AssemblyAI_LibriSpeech		Whisper_CommonVoice		AssemblyAI_CommonVoice	
	AUC score	Avg. Time	AUC score	Avg. Time	AUC score	Avg. Time	AUC score	Avg. Time
Temporal Dependency	0.5859	3.9500	0.4968	31.7900	0.5450	3.4800	0.5447	33.3800
Down-sampling & Up-sampling	0.6303	4.0500	0.7800	39.1000	0.6679	3.2900	0.6328	36.0000
Quantization & Dequantization	0.5329	4.0960	0.6483	38.0400	0.6477	3.3400	0.5991	33.7100
Filtering	0.6102	4.2800	0.7067	32.0600	0.6308	3.4200	0.6476	32.2000
Our detector	**0.9966**	**0.1250**	**0.9536**	**0.0120**	**0.9255**	**0.0190**	**0.8530**	**0.0150**

our detection mechanism is transferable across different datasets and STT models.

Table 6: Detection results II. AUC score, accuracy, precision, recall, and F1 score results of our detector under four different test scenarios.

Attacked Model	Dataset	AUC Score	Accuracy	Precision	Recall	F1 Score
whisper	librispeech	0.9966	0.9781	0.9899	0.9665	0.9780
assembly	librispeech	0.9536	0.9233	0.9379	0.9067	0.9220
whisper	commonvoice	0.9255	0.7667	0.7083	0.9067	0.7953
assembly	commonvoice	0.8289	0.7300	0.6994	0.8067	0.7492

The performance of our proposed detector for the four different scenarios is shown in Table 7. In those results, most of the points fall under correctly detecting clean or attacked audio samples. This suggests that our classifier is successful in identifying attacked audio samples. Moreover, we conducted more experiments on the detector in noisy environments, and the results are analyzed in Appendix A.2.1.

6.4 Regularized Robust Whisper

Figure 2 illustrates the finalized approach of our end-to-end defence, Regularized Robust Whisper. First, the detector classifies the input audio signal as attacked or benign. Only the attacked audio samples

Table 7: Confusion Matrix results of four scenarios.

True label Predicted label	clean attacked	attacked attacked	clean clean	attacked clean
Whisper_Librispeech	18	1757	1764	61
Assembly_Librispeech	9	136	141	14
Whisper_Commonvoice	56	136	94	14
Assembly_Commonvoice	55	123	95	27

are fed to Robust Whisper. If not attacked, we can use the already available Vanilla Whisper model for transcription.

Results of the Regularized Robust Whisper are shown in Table 8. We see that WERs of benign audio samples are reduced in the Regularized Robust Whisper in comparison to the standalone Robust Whisper in all four scenarios. We need to be mindful of the adversarial audio samples that mistakenly get detected as benign. These incorrectly identified attacked samples will be sent through the undefended Whisper, causing a higher WER. This is reflected in our results. When evaluated on attacked samples, you can observe a slight increase in the WERs in three out of four scenarios. But this increase is negligible compared to the improvement we achieve with the benign samples. Thus, the Regularized Robust Whisper performs better than the standalone Robust Whisper. We want to note that the zero WER achieved by the Defended Regularized solution for the benign samples in the LibriSpeech-AssemblyAI scenario

Table 8: Results of the Regularized Robust Whisper. The *Attacked Undefended* shows the WERs resulting from the undefended whisper model against attacked audios. It is used as a baseline to get an idea of the worst-case WER of each scenario. The analysis of the defended system is again divided into two sub-scenarios, (i) the performance with attacked samples (*Attacked Defended*) (ii) the performance with benign samples (*Benign Defended*). In each sub-scenario, we compare the results of the standalone Robust whisper and the Regularized Robust Whisper.

Dataset	LibriSpeech Whisper	CommonVoice Whisper	LibriSpeech AssemblyAI	CommonVoice AssemblyAI
Attacked Undefended	0.1042	0.4370	0.3535	0.8282
Attacked Defended Standalone	0.0730	**0.3630**	**0.3002**	0.5937
Attacked Defended Regularized	**0.0729**	0.3638	0.3009	**0.5903**
Benign Defended Standalone	0.0167	0.1453	0.0273	0.0790
Benign Defended Regularized	**0.0058**	**0.0589**	**0.0000**	**0.0422**

is caused by the number of test samples we used. We tested 150 benign audio samples to calculate it. All 150 were correctly identified by our detector, resulting in a zero WER. Further, we conduct more experiments in noisy environments and discuss the results in Appendix A.2.2.

6.5 Adaptive Attacks

In this paper, our goal is to design a defence against signal processing attacks. To the best of our knowledge, this is the first such defence. So it remains a future research problem to design and analyse possible adaptive attacks against these kinds of defences. To maximize the chance that this defence is robust, we proposed a new signal processing attack (Imaginary Clipping) and show our defence is robust to it. We leave for future work a full investigation of adaptive attacks. That said, it is worth discussing the limitations and advantages of our method concerning potential (future) adaptive attacks. As we have shown experimentally, our method is robust to noise, thus adaptive attacks that seek to add noise to evade detection are not expected to be successful. However, since our defence uses an underlying STT model, an attacker could target this model with an optimization-based attack, but our main focus in this paper is on signal processing attacks. Ultimately our argument is empirical in nature, but our methodology is in line with existing prior research, which often evaluates robustness against adaptive attacks [32] through a series of experiments. For example, Wang et al. [32] proposed a dynamic inference-time defence robust to adaptive attacks that optimize the model by minimizing the model output entropy. They validate the robustness of the defence through extensive experimentation on multiple models, datasets, and experimental settings. We followed a similar approach in our experiment to ensure the robustness of our defence by experimenting on an out-of-distribution dataset, multiple models and different levels of noisy environments.

7 CONCLUSION

In this paper, we propose a self-supervised fine-tuning algorithm — Robust Whisper, to make existing ASR systems robust to new adversarial attacks by filtering adversarial audio. Our approach has the advantage that it only requires the fine-tuning of the encoder of the STT model and does not require any human-annotated samples. We perform experiments across four scenarios, compare them with other state-of-the-art defence mechanisms and show that our approach performs better than existing methods. Robust Whisper used on both attacked and benign samples can result in transcription errors on benign samples. We elaborate on how combining Robust Whisper with a signal processing attack detection model, which detects attacked audio, can increase the accuracy of the overall approach. However, we believe that this problem can be handled by enough training data. We leave this as an open research problem for the future.

ACKNOWLEDGMENTS

The authors wish to thank our shepherd and anonymous AsiaCCS reviewers for their helpful comments in improving this paper. GPUs and the server used for this research are provided by Dr. Ranga Rodrigo at the Department of Electronic and Telecommunication of the University of Moratuwa, Sri Lanka. This work was supported by the Senate Research Council of the University of Moratuwa, Sri Lanka. This research was also supported in part by the U.S. National Science Foundation under CNS-1933208. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] 2023. Conformer-2: a state-of-the-art speech recognition model trained on 1.1M hours of data. https://www.assemblyai.com/blog/conformer-2/. (Accessed on 08/09/2023).

[2] Sajjad Abdoli, Luiz G Hafemann, Jerome Rony, Ismail Ben Ayed, Patrick Cardinal, and Alessandro L Koerich. 2019. Universal adversarial audio perturbations. *arXiv preprint arXiv:1908.03173* (2019).

[3] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin RB Butler, and Joseph Wilson. 2019. Practical hidden voice attacks against speech and speaker recognition systems. *arXiv preprint arXiv:1904.05734* (2019).

[4] Hadi Abdullah, Aditya Karlekar, Vincent Bindschaedler, and Patrick Traynor. 2021. Demystifying limited adversarial transferability in automatic speech recognition systems. In *International Conference on Learning Representations (ICLR)*.

[5] Hadi Abdullah, Aditya Karlekar, Saurabh Prasad, Muhammad Sajidur Rahman, Logan Blue, Luke A. Bauer, Vincent Bindschaedler, and Patrick Traynor. 2022. Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems. *arXiv preprint arXiv:2203.05408* (2022).

[6] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. 2021. "Hear" no evil", see" kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 712–729.

[7] Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. 2021. Beyond l_p clipping: Equalization based psychoacoustic attacks against asrs. In *Asian Conference on Machine Learning*. PMLR, 672–688.

[8] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. 2021. Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. In *2021 IEEE symposium on security and privacy (SP)*. IEEE, 730–747.

[9] Defossez Alexandre, Synnaeve Gabriel, and Adi Yossi. 2020. Real time speech enhancement in the waveform domain. *arXiv preprint arXiv:2006.12847* (2020).

[10] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. 2018. Did you hear that? adversarial examples against automatic speech recognition. *arXiv preprint arXiv:1801.00554* (2018).

[11] Sreeram Anirudh, Mehlman Nicholas, Peri Raghuveer, Knox Dillon, and Narayanan Shrikanth. 2021. Perceptual-based deep-learning denoiser as a defense against adversarial attacks on ASR systems. *arXiv preprint arXiv:2107.05222* (2021).

[12] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670* (2019).

[13] Nicholas Carlini and David Wagner. 2018. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. *arXiv:1801.01944v2 [cs.LG] 30 Mar 2018* (2018).

[14] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. {Devil's} Whisper: A General Approach for Physical Adversarial Attacks against Commercial Black-box Speech Recognition Devices. In *29th USENIX Security Symposium (USENIX Security 20)*. 2667–2684.

[15] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. 2017. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. *Advances in neural information processing systems* 30 (2017).

[16] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. 2019. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254* (2019).

[17] Thorsten Eisenhofer, Lea Schönherr, Joel Frank, Lars Speckemeier, Dorothea Kolossa, and Thorsten Holz. 2021. Dompteur: Taming audio adversarial examples. In *30th USENIX Security Symposium (USENIX Security 21)*. 2309–2326.

[18] Szu-Wei Fu, Cheng Yu, Tsun-An Hsieh, Peter Plantinga, Mirco Ravanelli, Xugang Lu, and Yu Tsao. 2021. Metricgan+: An improved version of metricgan for speech enhancement. *arXiv preprint arXiv:2104.03538* (2021).

[19] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2021. WaveGuard: Understanding and mitigating audio adversarial examples. In *30th USENIX Security Symposium (USENIX Security 21)*. 2273–2290.

[20] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. 2018. Fooling end-to-end speaker verification with adversarial examples. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 1962–1966.

[21] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[23] Raphael Olivier and Bhiksha Raj. 2021. Sequential randomized smoothing for adversarially robust speech recognition. *arXiv preprint arXiv:2112.03000* (2021).

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[25] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*. PMLR, 5231–5240.

[26] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*. PMLR, 28492–28518.

[27] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2018. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv preprint arXiv:1808.05665* (2018).

[28] Liu Andy T., Li Shang-Wen, and Lee Hung-yi. 2021. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio Speech Language Processing* 29 (2021), 2351–2366.

[29] Liu Andy T., Yang Shu-wen, Chi Po-Han, Hsu Po-chun, and Lee Hung-yi. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP*. IEEE, Barcelona, 6419–6423.

[30] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. 2019. Targeted adversarial examples for black box audio systems. In *2019 IEEE security and privacy workshops (SPW)*. IEEE, 15–20.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, Vol. 30.

[32] Dequan Wang, An Ju, Evan Shelhamer, David Wagner, and Trevor Darrell. 2021. Fighting gradients with gradients: Dynamic defenses against adversarial attacks. *arXiv preprint arXiv:2105.08714* (2021).

[33] Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*. PMLR, 5286–5295.

[34] Hiromu Yakura and Jun Sakuma. 2018. Robust audio adversarial example for a physical attack. *arXiv preprint arXiv:1810.11793* (2018).

[35] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. 2018. {CommanderSong}: A Systematic Approach for Practical Adversarial Voice Recognition. In *27th USENIX security symposium (USENIX security 18)*. 49–64.

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

[37] Zhao Zhengli, Dua Dheeru, and Singh Sameer. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342* (2017).

A APPENDIX

A.1 Human-Evaluation of Imperceptibility Survey

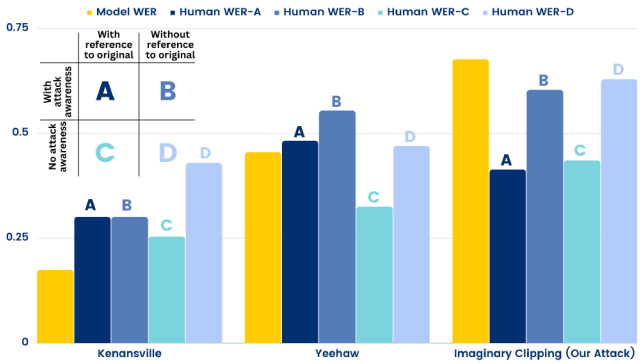

Figure 8: Model WER and human WER analysis of Kenansville, Yeehaw, and Imaginary Clipping (new attack) on four scenarios: A, B, C, and D. The conducted survey was to evaluate the imperceptibility and efficacy of signal processing attacks. The WER analysis was conducted based on the model and human-generated transcriptions. Results depict the complexity of achieving the imperceptibility level and show the competitiveness of the Imaginary Clipping attack.

Being imperceptible to the human ear is one of the main features of signal processing attacks. To validate the assumption, we conducted a survey involving 100 participants. The survey focused on evaluating the imperceptibility of signal processing attacks and their efficacy. To explore various scenarios comprehensively, we divided the participants into four distinct groups. Initially, they were randomly assigned to two groups based on their awareness of signal processing attacks; one group was informed about the nature of these attacks, while the other remained unaware. Subsequently, each of these two categories was further subdivided into two additional groups based on whether they had access to the original audio samples or not. One critical question to the user was to listen to the provided audio files and write what they heard. Using the resulting human transcriptions, a WER analysis was conducted. The model WER and the human WER with respect to the original transcription for each attack type were determined using the analysis. Figure 8 shows the obtained results.

The results show that when the audio sample is attacked with the Kenansville attack, the model's WER was lower than all the scenarios in human WER. This suggests that human listeners often misidentified the transcriptions when exposed to the attacked audio. In an ideal attack scenario, the human WER should be lower than the model WER while remaining imperceptible to the human ear, as this indicates that the model can be deceived without being detectable to the human ear. However, the results indicate that the Kenansville attack failed to fully maintain this goal, as the human WER was higher than the model WER. Similarly, the Yeehaw Junction attack showed that the WER for humans was higher than the model WER in most cases. Again, this contradicts the conditions for an ideal attack scenario. In contrast, the new attack demonstrated that the human WER was lower than the model WER in every case. But the human WER of the new attack is higher than the human WERs of the other two attacks. This indicates that the new attack has a high probability of being detected compared to the other two attacks. However, the results in Table 9 support the idea that the new attack demonstrates competitive performance compared to the state-of-the-art Yeehaw Junction attack in terms of imperceptibility.

Table 9: Ability of survey participants to identify attacked audio for each attack.

Attack	AUC Score	Accuracy	Precision	Recall	F1 Score
Kenansville	0.6840	0.6164	0.8947	0.4811	0.6258
Yeehaw Junction	0.7945	0.7687	0.9167	0.7021	0.7952
Imaginary Clipping	0.8074	0.7753	0.9381	0.7280	0.8198

In the survey, we prompted the users to rate their confidence in identifying an attacked audio. Table 9 depicts the results, which show how well the survey participants are capable of detecting an attacked audio. The considered metrics are Area Under Curve (AUC) score, accuracy, precision, recall, and F1 score values for Kenansville, Yeehaw Junction, and Imaginary Clipping attacks. Thus lesser the metric values, the participants find it difficult to recognise an attacked audio. So, the attack with lesser values is better in terms of achieving imperceptibility. Using the obtained results, it is clear that these attacks can achieve the state of imperceptibility only to a certain level. From the observations, we can arrive at the conclusion that the Kenansville attack has a better imperceptibility level than the other two attacks and the new attack demonstrates performance on par with the Yeehaw Junction attack, showcasing comparable results.

As a final note, we want to highlight that the proposed new attack contributes to making our defence more robust. The new attack helps to generalize the distribution of attack data that we use for training and evaluation of our models.

A.2 Experiments in Noisy Environments

A.2.1 Detection in Noisy Environments. One important factor to consider while building a benign sample detector is its behaviour in a noisy environment. We want to avoid benign samples that are coming from noisy environments being flagged as attacked samples. As a responsible security application, the detector should be able to notify the user about signal processing attacks with high confidence. To ensure this we tested our detector on benign samples coming from different noisy environments. For the tests, we used a segment of the Common Voice [12] dataset and manually added different levels of Additive White Gaussian Noise (AWGN) by controlling the SNR to simulate the noisy environment. We named this new dataset, which has different noise levels, as *cl_awgn_commonvoice*. It contains 80 audio samples per SNR level (0, 5, 10, 20, 30, clean). Table 10 shows the results of our tests. It shows a high accuracy at high SNR levels and low SNR levels. However, the model gets confused at mid-SNR levels. The main reason is the similarities in the spectral domain. AWGN noise adds the same level of amplitudes across all frequencies, resulting in flat frequency domain patterns.

As we explained in Section 2.3, the decimation and clipping in signal processing attacks result in similar flat frequency domain patterns. Also, the Yeehaw Junction attack [5] adds noise while querying the STT model. This is also a cause of this ambiguity.

As a summary, we have tested our benign detection model with three different types of audio samples — (i) clean benign samples, (ii) noisy benign samples, and (iii) attacked samples. As we discussed in Section 4.3, we want our detector to only feed attacked samples to the Robust Whisper to reduce errors introduced by our system. So, our detector needs to classify both clean benign samples and noisy benign samples as benign. Results of Section 6.3 and Section A.2.1 show the ability of our detection model to achieve this behaviour.

Table 10: Accuracy of correctly identifying benign samples under noisy environments

SNR (dB)	0	5	10	20	30	Clean
Accuracy	1	1	0.975	0.9375	0.6375	0.8

A.2.2 Regularized Robust Whisper in Noisy Environments. Our main goal is accurately identifying signal processing attacks and giving robust transcriptions for them. As we discussed in Section A.2.1, our detection model is able to classify benign samples coming from noisy environments as benign samples. So our end-to-end system, Regularized Robust Whisper, will direct them to the Vanilla Whisper model. So it is important to note that transcription errors the Vanilla Whisper model causes in noisy environments will appear in our end-to-end solution. We did tests to ensure this. Similar to the experiment in Section A.2.1 we picked 250 benign samples from a segment of Common Voice [12] dataset and added different levels of AWGN, Dataset has 50 benign samples per SNR level (0, 5, 10, 20, 30). We named this new dataset, which has different noise levels, as *ae_awgn_commonvoice*. Then, we calculated the WERs of the transcriptions given by the Vanilla Whisper STT model and Regularized Robust Whisper. Table 11 shows the results of our tests. The WERs made by the two systems are the same. This ensures that our detector has accurately identified benign samples and directed all of them to Vanilla Whisper. Even though a robust system should be robust to any kind of distortion, since our goal in this paper is building a defence against signal processing attacks, we leave this problem for future research.

Table 11: Regularized Robust Whisper behaviour against benign samples in noisy environments. *Benign Undefended* shows the WER Vanilla Whisper STT model makes for noisy benign samples. *Benign Defended Regularized* row shows the WER Regularized Robust Whisper makes for noisy benign samples.

SNR(db)	0	5	10	20	30
Benign Undefended	0.4298	0.9016	0.3717	0.2231	0.1440
Benign Defended Regularized	0.4298	0.9016	0.3717	0.2231	0.1440