IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 April 2024, accepted 15 June 2024, date of publication 20 June 2024, date of current version 1 July 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3416965*

SURVEY

# Adversarial Attacks on Automatic Speech Recognition (ASR): A Survey

**AMISHA RAJNIKANT BHANUSHALI, HYUNJUN MUN, AND JOOBEOM YUN**
Department of Computer and Information Security and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea

Corresponding author: Joobeom Yun (jbyun@sejong.ac.kr)

This work was supported by the Ministry of Science and Information and Communication Technology (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program, supervised by the Institute for Information and Communications Technology (ICT) Planning and Evaluation (IITP) under Grant IITP-2023-2018-0-01423.

**ABSTRACT** Automatic Speech Recognition (ASR) systems have improved and eased how humans interact with devices. ASR system converts an acoustic waveform into the relevant text form. Modern ASR inculcates deep neural networks (DNNs) to provide faster and better results. As the use of DNN continues to expand, there is a need for examination against various adversarial attacks. Adversarial attacks are synthetic samples crafted carefully by adding particular noise to legitimate examples. They are imperceptible, yet they prove catastrophic to DNNs. Recently, adversarial attacks on ASRs have increased but previous surveys lack generalization of the different methods used for attacking ASR, and the scope of the study is narrowed to a particular application, making it difficult to determine the relationships and trade-offs between the attack techniques. Therefore, this survey provides a taxonomy illustrating the classification of the adversarial attacks on ASR based on their characteristics and behavior. Additionally, we have analyzed the existing methods for generating adversarial attacks and presented their comparative analysis. We have clearly drawn the outline to indicate the efficiency of the adversarial techniques, and based on the lacunae found in the existing studies, we have stated the future scope.

**INDEX TERMS** Adversarial attacks, adversarial samples, automatic speech recognition (ASR), deep neural network (DNN).

## I. INTRODUCTION

In the real world, voice is the crucial medium through which humans communicate, thus sharing ideas, emotions, and our identity. The voice is the speaker's signature, which gets fabricated into speech. In the last few decades, with advancements in technology and the availability of computational capacity, speech has been widely accepted in devices, resulting in technical breakthroughs. The dependency on the speech command is increasing as it eases the operating process of real-time applications. Incorporating speech into devices has solved various crucial problems; as on devices where we cannot accommodate hardware, such as keyboards, speech becomes a reliable means of input for interacting with those devices; it relieves the users from the cumbersome task of typing. Today, Automatic Speech Recognition (ASR) has gradually progressed from traditional ASR into modern deep-learning-based ASRs. ASR accepts raw waveform as input speech, translating it into its corresponding text with minimal or null error rate. Numerous ASR applications such as Apple Siri, Amazon Echo, Google Assistant, and Google Home [1] depend on Artificial Intelligence (AI) techniques.

ASR has medical usage where a paralyzed person drives a wheelchair just by giving speech instructions; speech instructions are used for flying autonomous vehicles. Additionally, ASR is used in autonomous driving vehicles to assist users in communicating with the car controls and navigation system. According to Google's report [2], ASR was primarily used amongst groups of friends for communicating and performing tasks such as cooking, exercising, watching television, calling someone, asking for directions, helping with homework, playing a song, finding movie timings, and checking time. At the earliest, speech recognition was achieved by template matching, and with the development of Hidden Markov Models (HMM) in the 1970s [3], the performance of ASR increased substantially. AI mimics the

The associate editor coordinating the review of this manuscript and approving it for publication was Ganesh Naik.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

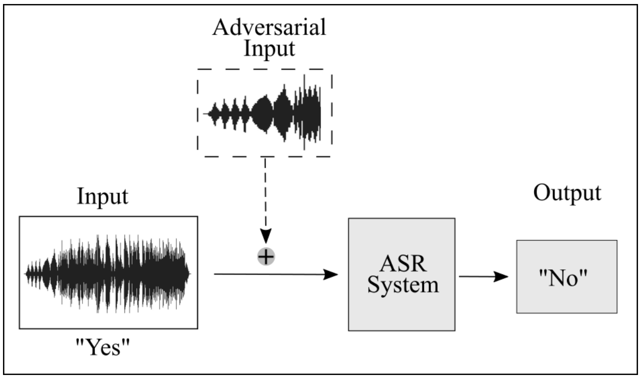


**FIGURE 1.** Adversarial attack on ASR.

same human intelligence capabilities for recognizing images, audio, and language. This is accomplished by DNNs, which are trained on standard datasets to perform speech-to-text translation. The DNN-based ASR was introduced in 2012, where only a small component of the ASR pipeline was integrated with deep learning models [4], [7]. Nowadays, deep learning methodologies have boosted the end-to-end approach in the ASR model [9].

With the invention of adversarial attacks (AA) in the image domain, the advancement of DNNs was pushed backward. It was observed that the Convolutional Neural Network (CNN) [15] can be deceived by the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [8] approach into predicting incorrect results just by perturbating the input pixel slightly. Similarly, researchers started exploring adversarial attacks within the audio domain. Initially, adversarial samples (AS) were generated by using the optimization method for the music genre classification task [10]. In contrast, adversaries were created for speech paralinguistics applications using the Fast Gradient Sign Method (FGSM) technique [11], [12]. They developed a strategy to explicitly disrupt the raw waveform rather than specific acoustic features, thus preventing the perceptual loss generated by translating acoustic data back to the waveform. As shown in Figure 1, an adversary harms the ASR by leading it to misinterpret the output text that the user initially desired. Adversarial attacks in audio can occur when playing sound that's recognized as something else, like speech being understood as different words, music being mistaken for a command, or hiding speech using psychoacoustic methods [13].

However, AAs on audio recognition are difficult to generate because of the structure of ASR. The speech signal must be pre-processed and transformed from the time domain to the frequency domain. Additionally, there are many languages, each with its unique vocabulary and pronunciation, which adds to the complexity of the issues that must be solved [14]. Audio recognition plays a vital role in our lives, and automatic audio recognition technology, along with image and text recognition, has advanced dramatically. We rely on ASR devices for day-to-day tasks such as sending text messages, calling, and making bank payments. With voice replacing text, attackers can discover an ASR system's vulnerability. There are many malicious users, and they might want to control our smart devices without us being completely aware of it.

Our inspiration for this study was to develop a taxonomy clearly classifying the AAs in ASR and describing their characteristics. *What distinguishes the proposed study from earlier survey papers?* All of the currently published survey papers consider limited scope while analyzing the AAs in ASR [16], [17] and do not assess how well these methods perform to clearly distinguish the advantages and disadvantages of various research. This is the first survey paper to illustrate the chronological advancement in the AA on ASR, thus classifying the methods and highlighting a detailed taxonomy based on their characteristics. *What are the various strategies that affect how well an AA method works?* The existing survey papers fail to answer this question [18], [19]. The AA methods are made up of various processes, including optimization (opt), gradient estimation (GE), and interpolation; these techniques have a direct impact on the AS that are produced. Thus, in this survey, we investigate the AAs on ASR and offer a taxonomy with a comprehensive review of the various techniques. This survey will help researchers and practitioners follow up on state-of-the-art AA methods on ASRs and give insights into the future scope.

We have summarized the paper in the following sections. Section II includes the survey method, and Section III introduces the background of ASR. The threat model and adversarial terminologies are explained in Section IV. Section V proposes a taxonomy of adversarial attacks on ASR, and Section VI provides a comparison and discussion of the methods used for generating adversarial attacks on ASR. Section VII highlights the future direction and finally, the article is concluded in Section VIII.

## II. SURVEY METHOD

We meticulously obeyed the guidelines given by Kitchenham [33] and Webster & Watson [34] for conducting a comprehensive survey of adversarial attacks on ASR. We followed a systematic approach by providing a structured literature review and have ensured thoroughness and consistency in our review process. Moreover, we selected papers from top journals and conferences to minimize bias leading to more reliable results. Our analysis of the various adversarial methods will assist future researchers to refer our work and develop various other adversarial techniques or defense mechanisms. The quality of the research findings has given more importance which will help future researchers to gain insights from this survey. In the given section, we have outlined the research queries, paper selection and exclusion strategy, and collection summary.

### A. RESEARCH QUERIES

The following research questions are addressed in our work.

1) What is the difference between traditional ASR and modern ASR?

2) How are adversarial attacks categorized in the audio domain?
3) Why is it difficult to generate adversarial attacks on ASR?
4) What is the current research analysis and future direction?

The answers with respect to the research queries are listed below. The background of both modern and traditional ASR is provided in Section III-A and III-B as an answer to query 1. Section V delivers the answer to query 2 inspiring us to investigate attacks on ASR and a taxonomy is provided depicting a clear classification of AA on ASR. A detailed literature study about the method for generating AA in the audio domain is presented in Section V-B stating the answer to query 3. Finally, query 4 is addressed by revealing the comparison and future scope in Sections VI and VII.

### B. PAPER SELECTION AND EXCLUSION STRATEGY

We collected and carefully curated a wide range of relevant articles in order to perform a thorough survey based on the ASR applications, methods, tools, usage properties, evaluation metrics, and existing surveys. The collection strategy included the following steps:

- Search the keyword "ASR", adversarial attack on ASR, adversarial attack on audio, adversarial attack on speech in Google Scholar, and filter according to the year.
- We have considered the papers that are written in English.
- Select papers with more than 5 pages in length.
- Include papers that are accessible over the internet.

In our survey, we constructed a repository of all the relevant studies to cover a broader aspect of the publications considering the ASR. The papers that we have included in our database followed the following rules: Firstly, we gathered papers from the year 2015 till December 2023. Secondly, we have analyzed more than 200 articles, from which we have selected 65 research articles providing a thorough literature review focusing on ASR. Finally, we collected and categorized the papers related to the following libraries: IEEE Xplore [35], ACM Digital [36], Springer [37], and Elsevier ScienceDirect [38]. We chose the papers based on the aforementioned criteria and collection strategy.

Initially, we have just read the abstract, the discussion, and the conclusion section of the paper to check the relevance. If the paper is relevant, we thoroughly read it to state the taxonomy and classify the studies based on their features. Thus, from the filtered 200 studies, we have only considered the most significant 65 studies. Table 1 presents the most relevant studies retrieved from each library.

### C. COLLECTION SUMMARY

In this section, we have highlighted the relevant studies based on libraries and publication terms.

As observed in Table 1, we only gathered the papers from prestigious publications and the most cited libraries. However, we ensure that our rigorous strategy accurately maps the relevant state-of-the-art studies considering the AA in ASR. Figure 2 represents the papers published on adversarial attacks on ASR between the years 2015 and 2023. AAs were discovered in 2013 in the image domain and they were explored in the audio domain in the year 2015 which itself explains the shortcomings of the audio domain. As depicted in the graph, there has been minimal growth in the number of papers from 2015 to 2016, while the number twofold in 2017. With the broader scope of applications in ASR, the number of publications increased exponentially from 2018 to 2020. There has been a steady growth in published papers from 2021 to 2023. If this trend continues, there are likely to be more research articles on this subject.

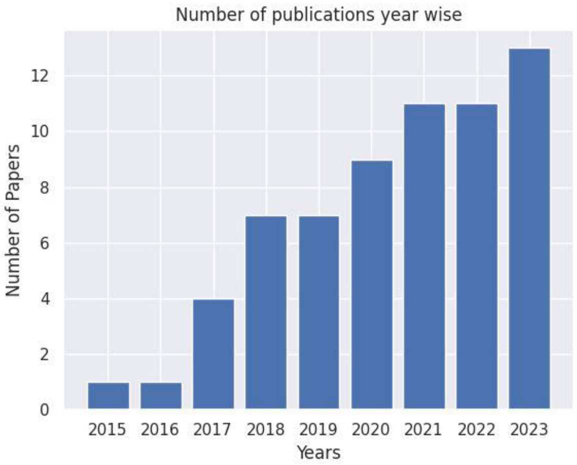


**FIGURE 2.** Number of papers published between 2015 and 2023.

**TABLE 1.** Libraries and the number of relevant studies.

| Publisher | Relevant studies |
|---|---|
| ACM digital library | 9 |
| Elsevier ScienceDirect | 3 |
| IEEE Xplore digital library | 21 |
| Springer online library | 6 |
| MDPI | 1 |
| USENIX | 6 |
| arXiv | 16 |
| Others | 3 |
| Total | 65 |

Figure 3 represents the chronological overview of adversarial attacks on ASR. We classified the studies and clustered them to draw a significant difference based on their methods for generating AA. The green arrows represent optimization attacks, which come under the white-box. It is evident from the figure that most of the papers used white-box as their threat model followed by black and gray-box. As commercial ASRs (C-ASR) are proprietary and only come with little public information, black-box attacks play an essential role in the real world. In the early years (2015-2019), most papers attacked traditional or hybrid ASR models by manipulating the MFCC features or changing

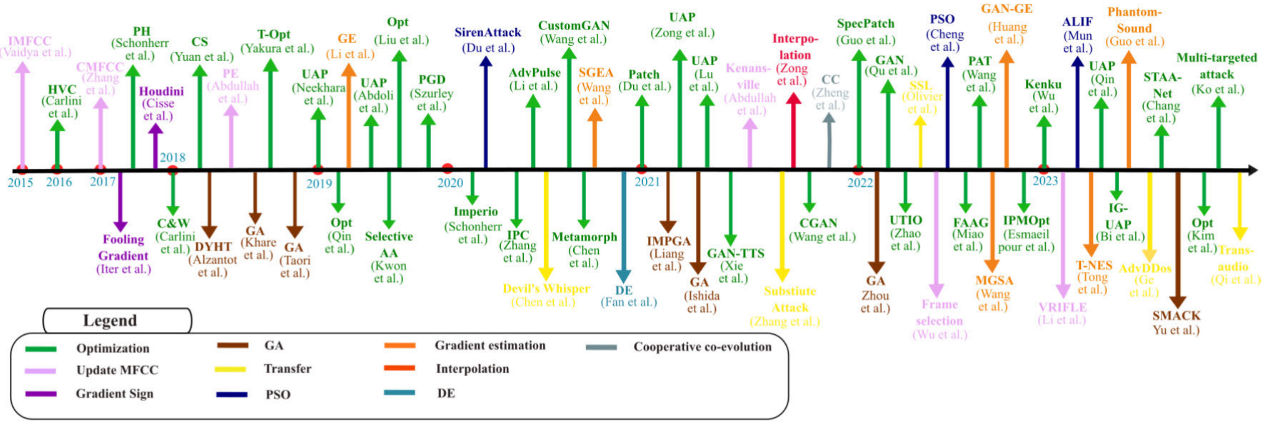


**FIGURE 3.** Chronological overview of adversarial attacks on ASR.

the gradients. These methods considered the pre-processing steps and relied on signal-processing techniques for attacking the ASR. Later studies indulged in attacking end-to-end modern ASRs that used DNNs for processing speech. They concentrated on attacking the DNN model using optimization (opt), transfer (TA), interpolation, gradient estimation (GE), and evolutionary methods. Evolutionary algorithms such as Genetic Algorithm (GA) [99], Particle Swarm Optimization (PSO) [111], Cooperative Coevolution (CC) [115], and Differential Evolution (DE) [112] were the second most used methodologies for crafting AS.

## III. AUTOMATIC SPEECH RECOGNITION

This section discusses the evolution of the ASR from a traditional to a modern ASR and provides the answer to the research query (1). Figure 4 gives a detailed depiction of the traditional ASR vs modern ASR system.

### A. TRADITIONAL ASR

The concept of creating machines that could understand and transcribe spoken language emerged in the 1940s and 1950s and since then, efforts have been undertaken to continuously improve the ASR model. Earlier traditional systems mainly relied on isolated word identification and employed simple signal-processing techniques and pattern-recognition algorithms. The traditional ASR system primarily consists of four stages as depicted in Figure 4. The first stage is feature extraction, followed by acoustic, pronunciation, and language models.

First, the unwanted noise that is inaudible to humans is removed with the help of low-pass filters before sending the speech signal into the system. The filtered signal is divided into overlapping frames, usually 20 ms long, and passed into the feature extraction stage for deriving the necessary acoustic feature vector. The feature vector represents all the information in the signal. There are various feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCC) [20], Linear Predictive Coefficient (LPC) [23], and Perceptual Linear Predictive (PLP) [22], amongst which MFCC is primarily used in commercial devices and toolkits. The MFCC is obtained by first converting the signal from the time domain to the frequency domain with the help of Discrete Cosine Transform (DCT), and generating a spectrogram. Because humans recognize low frequencies better than high frequencies andÂ particularly perceive frequency logarithmically, the spectrogram is transformed into a mel spectrogram. To reduce the correlation between variables, the last step transforms the highly correlated frequency domain information into a new domain by using DCT [24].

Second, after getting the MFCC feature vector, the acoustic model expresses the relationship between extracted feature vectors and phonemes. In other words, the acoustic model finds the phonemes that best represent a given feature vector. A phoneme is a discrete and distinctive unit of language that is used to differentiate between words and represent sound. An acoustic model and pronunciation model comprised of the Gaussian Mixture Model (GMM) and HMM, where the GMM was incorporated for modeling the probability distribution based on the input feature vectors for associating the states in the HMM. The GMM increases the level of accuracy by fitting the data using the Estimation Algorithm [21] and befitting HMM to deal with the temporal variability of speech depending upon the probability. Thus, it helps HMM to align the sequence of speech by considering the phonemes in a specific order.

Third, after getting the probabilistic vector values from the acoustic model, the pronunciation model maps a word to the corresponding pronunciation of the word. The pronunciation model considers multiple acceptable pronunciations, dialects, and accents per word to recognize words accurately.

Finally, the language model learns the corresponding language's grammatical framework, such as lexical selection and sentence-wise syntactic structure. The language model improves the recognition rate by increasing the probability of grammatical sentences and contributes to speeding up audio recognition by filtering words that need not be explored in advance. N-gram [26] is commonly used at this stage, which defines the probability of the next word from n-1 past words.

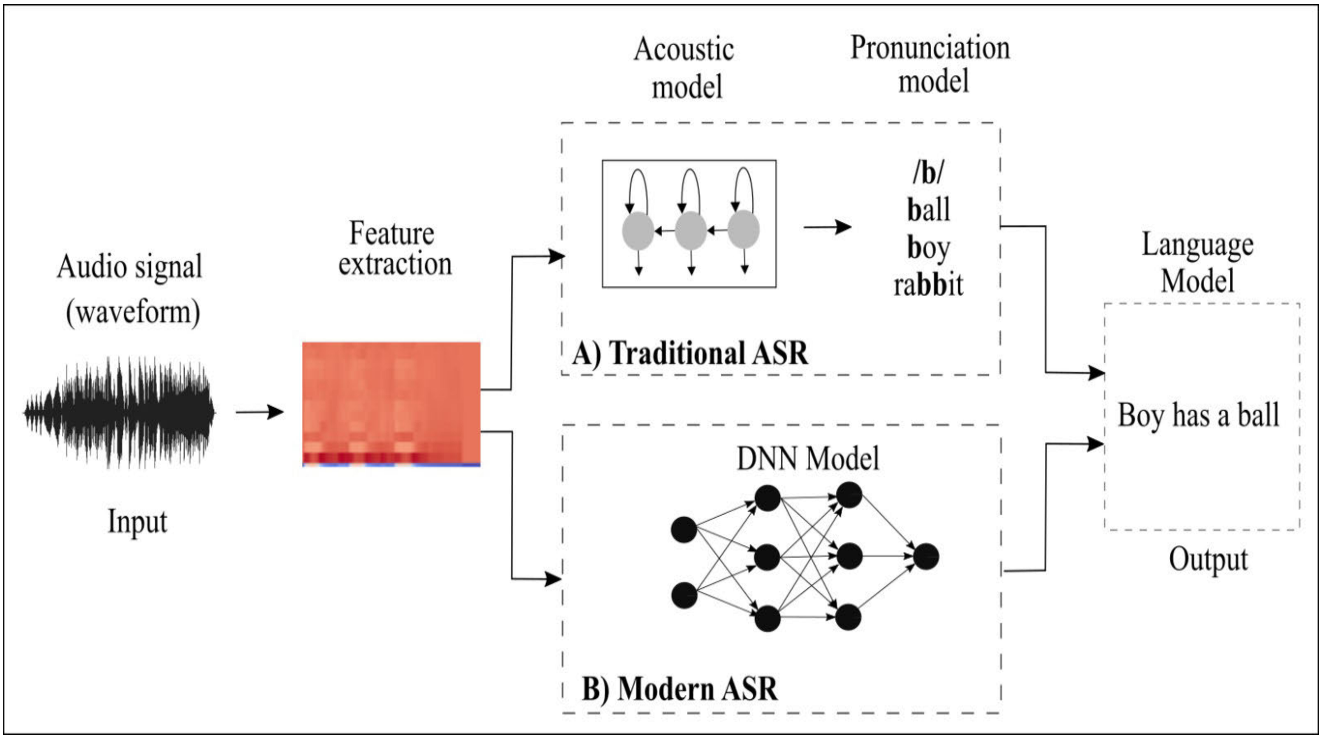


**FIGURE 4.** Traditional vs Modern ASR system.

Most traditional ASR highly depends on the corresponding stages for transcribing the audio.

### B. MODERN ASR

DNNs were a hotspot for researchers in 2010, and soon, they realized that each component of ASR seems to work better with a neural network. The traditional GMM-HMM architecture was replaced with DNN-HMM, making the ASR model hybrid. The main issue of the hybrid model was that each module had a different output, and it took a lot of work for the researchers to align every output from each module. Another problem is that the alignments derived from the HMM must be relied on to infer the frame-level training targets. HMM re-alignments are carried out to generate more accurate results, which is an unpleasant iterative process. Maximum Mutual Information, a full-sequence training method, was discovered to maximize the probability of developing more accurate results [25]. However, it was only suitable for retaining a system already trained at frame level. It also demanded a prudent tuning of hyper-parameters, typically even more than the tuning required for DNNs.

Modern ASR is an end-to-end speech recognition model and does not need any hand-designed components or phonemes. They directly map the acoustic input to text output without relying on intermediate phonetic or linguistic features. This simplifies the ASR pipeline and often results in improved performance. The primary motivation behind designing end-to-end models was to predict output sequences directly. Modern ASR system uses Connectionist Temporal Classification (CTC) [39] to align the next possible word in a sentence and maximize the likelihood of correct output given the input. The probability of any transcript is the sum of the probabilities of all paths that correspond to the transcript. We not only get the output probability accurately but also get the gradient. Once the gradient is obtained, we can backpropagate the Recurrent Neural Network (RNN) and learn the model's parameters.

### C. DIFFERENCE BETWEEN IMAGE DATA AND AUDIO DATA

This section discusses the difference between image data and audio data based on data processing and models. Table 2 illustrates the difference between audio and image data.

#### 1) DATA PROCESSING

Data processing in images and audio involves distinct methods and considerations due to the differences in the nature of the data. An image is something that can be viewed and is categorized under computer vision. Images are made up of pixels and contain one or more channels (i.e., Grayscale, RGB) resulting in 2D or 3D arrays. The number of pixels in an image is fixed while processing the image data. Audio, on the other hand, is heard and is covered by Natural Language Processing (NLP). Audio is represented in the form of waveform consisting of frequency and amplitude at a specific time resulting in 1D data [69]. Images are static and directly processed by a DNN, whereas audio is continuous and requires a lot of pre-processing. As the length of the output in audio is variable and has exponentially many labels,

**TABLE 2.** Difference between image and audio data.

| | Audio | Image |
|---|---|---|
| Data Representation | 1D Data | 2D Data |
| Type | Sequential | Discrete |
| Input Data Size | Variable length | Fixed length |
| Preprocessing | Computationally expensive as it goes through conversion and rectification | Moderate as it goes through only data augmentation |
| Adversarial attack target | Waveform | Pixels |
| Visualization | Often confusing and hard to interpret | Easy to understand |
| Execution | Mostly offline | Both (online and offline) |
| Susceptible | Noise, Reverberations | Brightness, Fog |
| Extension | .wav | .png, .jpg |

it makes the audio processing computationally expensive compared to images.

### 2) MODEL

Deep learning models have significantly advanced handling both image and audio data, enabling various applications in computer vision and speech processing. However, not all models that work for images can be applied to audio. As the output of image data is fixed, CNNs are primarily used for detection, classification, segmentation, and action recognition [66], [67]. The convolution layers learn the features automatically, thereby producing accurate results. RNNs are used for sequential data like audio as input length is variable [92], [93]. CNNs are also used for audio-related tasks that involve sound classification [90], [157]. The models used for audio have to be trained with an enormous amount of data since each audio input can be of variable length. The dataset is large and, hence, very computationally expensive to train an audio model.

## IV. ADVERSARIAL ATTACK ON ASR

This section provides information on the threat model, perturbation, attack features, benchmark datasets, models, and metrics used for evaluating AA on ASR.

### 1) THREAT MODEL

The threat model describes the circumstances under which AAs are produced. It may be classified using a variety of criteria. Here, we concentrate on the threat model that is characterized by adversarial knowledge and specificity [27]:

- Adversarial knowledge
    1) White-box: The adversary has full knowledge of the model architecture, training data, and weights of the victim model. The adversary makes use of model information for generating AAs.
    2) Gray-box: The adversary is aware of the training data excluding the model architecture. As a result, the adversary can query a model and estimate the parameters to produce an AA.
    3) Black-box: The adversary is unaware of the settings of the victim model. He utilizes data gathered by examining and keeping track of the victim model's inputs and outputs.
- Adversarial Specificity
    1) Targeted: The adversary creates the adversarial attack to lead the ASR model astray and causes it to misinterpret the input sample to a particular target label $l'$. These attacks are very focused and are performed by increasing the target label's likelihood. Due to the limited area available to guide the adversarial attack to a target label, these attacks are more challenging to perform as compared to untargeted attacks. As a result, the targeted attacks have lower success rates than untargeted attacks.
    2) Untargeted: The adversary creates the adversarial attack in order to lead the ASR model astray and cause it to categorize the input sample into a target label $l'$ that is different from the proper label $l$. The attack is created by minimizing the likelihood that label $l$ is accurate. The attacks can be carried out theoretically by creating a number of targeted attacks and choosing the one that causes the least disruption.

### 2) PERTURBATION

- Perturbation scope
    1) Individual: The individual perturbation is generated differently for each original input. Each perturbation is optimized for each input value, and when applied to other inputs, it may result in the loss of adversarial features or unintended consequences [8].
    2) Universal: The universal perturbation is one perturbation that can be applied generally to the entire dataset. The concept of universal adversarial perturbation for the first time by mainly iteratively optimizing one perturbation for multiple

inputs [28]. Unlike individual perturbations that must be optimized for each input, pre-generated universal perturbations can consistently attack all inputs in real-time. So, universal perturbation is suitable for the real world.

- Perturbation measurement
  1) SNR: Signal-to-noise ratio is a metric that measures the ratio between the power of original audio $x$ and the power of perturbation $\delta$ [43]. The units of expression of SNR is dB, and is formulated as follows:

$$\mathrm{SNR}(x, \delta) = 20 \cdot \log_{10} \frac{P_x}{P_\delta} \tag{1}$$

  The smaller the perturbation compared to the audio, the larger the SNR. So the larger the SNR, the harder it is to notice the noise.

  2) $l_p - norm$: The $l_p - norm$ measures the distance between the original input and the adversarial example as p-norm [42]. The p-norm is formulated as follows:

$$||x||_p = \left( \sum_{i=1}^{n} ||x_i||^p \right)^{\frac{1}{p}} \tag{2}$$

  $l_2$ and $l_\infty$ norm are the most commonly used metrics in audio domains. $l_2$ is the Euclidean distance between the original input and the adversarial example, and $l_\infty$ is the maximum transformation.

  3) Power Spectral Density (PSD): PSD describes the distribution of power over frequency. It tells how strong a signal is [80]. Prominent frequencies in the signal are represented by peaks in the PSD graph.
  4) Levenshtein distance (LD): Levenshtein distance gives a value by comparing how similar the two characters are [41]. It computes the number of insertion, deletion, and substitution operations required to convert one string to another.
  5) Sound Pressure Level (SPL): SPL is the amount of pressure that sound waves exert when passing through a transmission medium. It is used to state the intensity of the sound [77]. It offers a standard logarithmic scale for quantifying sound intensity in a way that is in accordance with human perception.
  6) Total Variation Denoising (TVD): TVD is used to calculate the amount of noise present in a wave [106]. The higher the TVD, the more is the noise present. It is based on the idea that signals with excessive and potentially incorrect detail have a huge total variation.

### 3) ATTACK FEATURES

- Waveform: Waveform is the standard format used to represent the audio signal. It is made up of frequencies and amplitudes. As sound travels through air, it causes the air molecules to oscillate and the changes in air pressure creates a wave. Most AAs are carried out on the raw waveform.
- Spectrogram: Spectrogram is a high-dimensional heat map through which we can visualize audio. The color intensities in the spectrogram represent the volume of audio. The lower the pitch the lower the intensity in the graph, the higher the pitch the higher the intensity on the graph. The frequency scale is linear representing the distribution of frequency over time.
- MFCC: MFCC stands for Mel-Frequency Cepstrum Coefficients. The term Mel-Frequency represents the values from the Mel scale. Humans do not interpret sound linearly, instead, they perceive it logarithmically. Mel scale was invented as it is a perceptually relevant scale with respect to the human auditory system. Cepstrum on the other hand is a reverse of the spectrum developed in the 1960s for studying echoes in seismic signals [65]. MFCC looks like a matrix with proper sections of time data. It represents the data in a more compact way than a spectrogram.

### 4) OVER-THE-AIR

When the audio is played on a speaker and recorded by a microphone physically, the condition is said to be over-the-air [98]. Physical attacks in the image domain are more straightforward to carry out than in the audio domain since audio, when recorded, is influenced by various environmental factors, such as reverberations, noise from both the speaker and the microphone, and room arrangement. Generating AS that are robust over-the-air is challenging, considering unknown environments and equipment.

### 5) BENCHMARK

- Dataset: A variety of datasets are used for training ASR. The Speech Command dataset (SCD) [30] and Mozilla Common Voice dataset (MCVD) [31] are very popular. The SCD contains 65,000 utterances of audio data which are 1 sec long whereas MCVD contains more than 7000 hours of recorded audio data. Moreover, there are many datasets available that are designed to support different languages such as chinese, russian [63], and hindi [64]. AISHELL [59] is an open-source Mandarin corpus consisting of recordings of 400 people. Librispeech [29] consists of 1000 hours of English utterances and mini librispeech is a subset of it designed for the purpose of regression testing. The TIMIT dataset [144] consists of 6300 recordings of 3.14 hours. It includes phonetically rich phrases where 30% are female voices and the rest are male. The Wall Street Journal Dataset (WSJD) [66] is an English-speaking dataset containing 400 hours of speech and is mostly used for NLP tasks.
- Model: DeepSpeech [32] is the most widely used open-source ASR system created by Mozilla. It uses deep learning techniques, notably CNNs and RNNs [89] to translate spoken language into written text. Since

DeepSpeech is designed for real-time inference, it can be used for live captioning and other low-latency speech recognition applications. Keyword Spotting System (KWS) [90] or other CNN-based ASRs are used on a small scale to convert utterances into corresponding words. These systems typically use CNNs as one of the components for feature extraction. The input audio is processed through CNN layers to capture relevant acoustic features. This is often followed by recurrent layers (e.g., LSTM or GRU) [91] to model temporal dependencies and generate text transcriptions. The KWS is trained to recognize particular keywords and is often used in smartphones for command recognition. Kaldi [6] is another open-source toolkit that is used for speech recognition. It consists of various ASR models used for speech recognition. ESPnet [60] is an end-to-end speech processing toolbox that includes implementations of multiple ASR models. It supports deep learning-based end-to-end models as well as traditional hybrid systems.

#### 6) METRICS

Metrics are crucial when assessing an ASR model. It provides insight into the model's performance, enabling us to determine how well or poorly a model has performed depending on the data. Metrics are essential to ensure the model functions properly and ideally. Depending on the intended use, various evaluation metrics are available. Present-day methods evaluate and compare the effectiveness of their attack using Signal-to-noise ratio (SNR), Word Error Rate (WER), Success Rate (SR), and Number of queries.

- SNR: SNR is the measure of signal-to-noise ratio. A larger SNR indicates a smaller perturbation, meaning the generated perturbation is difficult to interpret by a human [43].
- WER: WER is defined as the number of errors divided by the total number of words. It is given by calculating the number of substitutions, deletions, and insertions per total length of the word [71]. This metric is based on LD which finds out the resemblance between two values. The lower the WER, the better the performance.

$$\mathrm{WER} = \frac{S+D+I}{T} \tag{3}$$

- SR: It is a very common evaluation metric for adversarial attacks [86]. The SR is the ratio of the number of adversarial samples created that achieve the target goal.

$$\mathrm{SR} = \frac{AS}{TS} \tag{4}$$

- Number of queries: Number of queries represents the total number of queries a model fires to generate the AS [109]. The time consumption and computational cost increases with the increasing number of queries.

## V. TAXONOMY OF ADVERSARIAL ATTACK ON ASR

In this section, we have provided a taxonomy of adversarial attacks on ASR with a detailed classification. This section provides answers to the research queries (2) and (3). Figure 5 displays the taxonomy of the adversarial attacks on ASR.

### A. CLASSIFICATION

We have considered the first criterion for the classification as the type of adversarial input to be processed by the ASR. It depends on a variety of factors. The adversarial input is said to be inaudible if it is beyond the human hearing range and is generally created with the help of signal processing techniques. Perturbated input is crafted with precision and a goal to lead the ASR model astray which can be susceptible to humans. The adversarial input crafted with the help of a generative model is synthetic and is created by accumulating random noise. It includes two sub-models: a generator and a discriminator. A generator generates the fake adversarial samples, whereas the discriminator distinguishes between the real and fake samples. The second criterion is the level of knowledge, categorized as white-box, gray-box, and black-box. The adversary may use a variety of approaches to produce AS, depending on their level of knowledge and access to the target model. Attacks are considered a white-box if an adversary can access the target model's parameters. An adversary's confidence will likely increase while constructing an attack in the white-box since he has optimized the adversarial sample directly with the target model. Due to a lack of information on the model's characteristics, attacks designed under the black-box have a relatively low SR. The awareness of the model's parameters in the gray-box is minimal, with access to only the model's output probabilities. Depending on the first and second criteria, we set the third criterion as the method for generating AA. The details are discussed in the following subsection.

#### 1) INAUDIBLE INPUT

Humans can hear the sound between the frequency range of 20 Hz to 20 kHz. Inaudible inputs are those inputs that fall outside the human hearing range. It is a branch that differs from perturbated inputs in terms of auditory and adversarial sample generation. While perturbated input retains the acoustic properties of the original audio, inaudible input can be heard as noise in the human ear or not at all. The attacker conceals the adversarial input so that it is inaudible to users but detected by the ASR. Many researchers have exploited this phenomenon and developed attacks based on hidden commands and psychoacoustic hiding [75], [76], [78]. It can be categorized into white-box, gray-box, and black-box and the methods used for generating AA are opt, GE, and updated MFCC. Table 3 compares and provides the analysis of adversarial attacks based on inaudible input.

#### 2) PERTURBATED INPUT

Perturbations are well-crafted adversarial inputs carefully chosen to achieve desired results [8]. They are not created randomly but have a specific mechanism behind them. The perturbations created to achieve a particular goal are targeted,

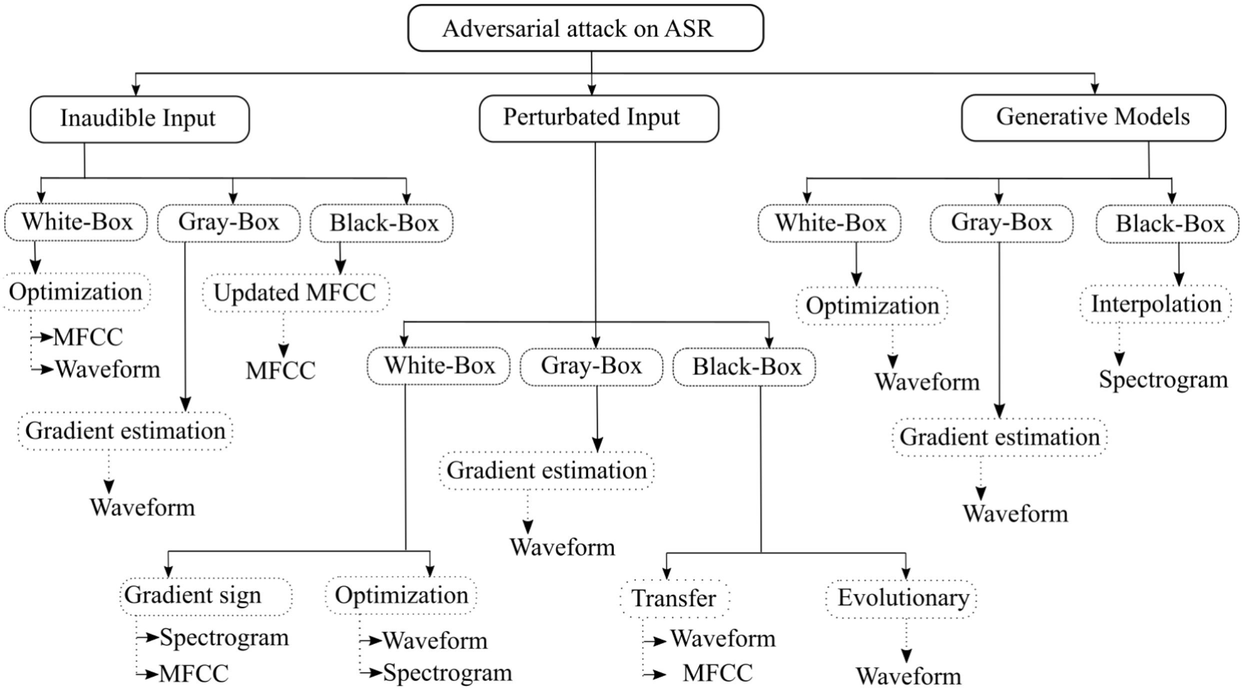


**FIGURE 5.** Taxonomy of adversarial attacks on ASR.

whereas perturbations designed just for changing the output are called untargeted perturbations. They can be individual or universal depending upon the scope. Perturbated inputs are categorized as white-box, gray-box, and black-box, and the methods for generating AA are the gradient sign, opt, GE, TA, and evolutionary. Tables 4 and 5 compare and provide the analysis of adversarial attacks based on perturbated input.

#### 3) GENERATIVE MODELS

Generative Models (GM) are the modern DNNs with two sub-models, a generator, and a discriminator [40]. The generator is designed to create the AS, and a discriminator must determine whether the samples are authentic. The adversarial input is random and crafted by combining various noises. A generator is trained to fool a discriminator into generating AS. Generators update themselves till they have created a convincing AS. They are categorized as white-box, gray-box, and black-box, and the methods used for generating AA are opt, GE, and interpolation. Table 6 compares and provides the analysis of adversarial attacks based on the generative model.

### B. METHODS FOR GENERATING ADVERSARIAL ATTACK ON ASR

This section discusses the details of the methods for generating adversarial attacks on ASR. The methods are as follows: Updated MFCC, Gradient-sign, Optimization, Gradient Estimation, Transfer, Evolutionary, and Interpolation.

#### 1) UPDATED MFCC

The MFCC features represent all the acoustic information carried by a waveform. Many researchers exploited the MFCC features to construct AS by simply updating them by changing or removing some values. Vaidya et al. [75] proposed an audio mangler algorithm that produced a morphed version of the original input. The audio mangler takes an audio input and updates the MFCC parameters so that some of the acoustic features are lost but present enough to be at least recognized by ASR. Afterward, the waveform is rebuilt using inverse MFCC; however, during this procedure, noise is introduced, which renders the waveform lossy and inaudible to humans. 1The experimental results show that the generated AS was inaudible to 95% of humans and appeared conspicuous to 5%.

Overcoming the above issue, Zhang et al. [77] made use of ultrasonic sounds that are beyond the human hearing range to deceive ASRs. By updating the extracted MFCC features and comparing them with target values, they could concatenate the features in the correct order and generate the voice commands. Similarly, Li et al. [86] designed a model to construct ultrasonic sounds for creating perturbations that can be physically effective and survive long distances. They proposed an alter and mute strategy to update the desired MFCC features. For the AS to survive in the real world, they used RIR and AIR [57]. They achieved an attack SR of 99.49% on the fluent speech dataset [58] by attacking DeepSpeech2, but their attack was device-specific.

Abdullah et al. [79] developed perturbations during the signal processing stage with the help of an AS generator called a Perturbation Engine. They devised four strategies to create perturbations: time domain inversion modifies the audio in the time domain before pre-processing, random phase generation selects any random number to update the

**TABLE 3.** Comparative analysis of adversarial attacks on ASR based on inaudible input.

| Year | Method | Target [1] Model | Target Dataset | Target [2] Object | Adversarial Method | Adversarial [3] Knowledge | Adversarial [4] Specificity | Perturbation [5] Scope | Perturbation Measurement | Over-the-air | Open Source |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2015 | Vaidya et al. [75] | GSA | N/A | MFCC | Update MFCC | BB | T | I | LD | N/A | Close |
| 2016 | Carlini et al. [76] | Sphnix,<br>C-ASRs | GMU | MFCC | Opt | WB<br>BB | T | I | $l_p$<br>SNR | 1-3m | Close |
| 2017 | Zhang et al. [77] | C-ASRs | N/A | MFCC | Update MFCC | BB | T | I | SPL | 0.75m | Close |
| 2017 | Schonherr et al. [78] | Kaldi | WSJD | WF | Opt | WB | T | I | SNR | N/A | Close |
| 2018 | Abdullah et al. [79] | ASRs | LS,<br>TIMIT | MFCC | Update MFCC | BB | T | I | $l_1$ | 0.34m | Close |
| 2019 | Qin et al. [80] | Lingvo | LS | WF | Opt | WB | T | I | PSD | N/A | Close |
| 2019 | Li et al. [81] | Alexa | Customized | WF | GE | GB | T | I | PSD | 2m | Close |
| 2019 | Szurley et al. [82] | DS | LS | WF | Opt | WB | T | I | $l_2$ | 6m | Close |
| 2020 | Schonherr et al. [83] | Kaldi | WSJD | WF | Opt | WB | T | I | SNR | 1-6m | Close |
| 2021 | Abdullah et al. [84] | DS,Sphinx,<br>C-ASRs | TIMIT | MFCC | Update MFCC | BB | UT | I | N/A | N/A | Close |
| 2022 | Wu et al. [85] | DS,<br>Sphinx | LS<br>MCVD | MFCC | Update MFCC | WB<br>BB | T<br>UT | I | PSD | N/A | Open |
| 2023 | Li et al. [86] | DS2 | Fluent Speech | MFCC | Update MFCC | BB | T | U | SNR | 1.8m | Close |

[1] Target Model (DS: DeepSpeech, DS2: DeepSpeech2)
[2] Target Object (WF: Waveform, SP: Spectrogram)
[3] Adversarial Knowledge (BB: Black-Box, WB: White-Box, GB: Gray-Box),
[4] Adversarial Specificity (T: Targeted, UT Untargeted),
[5] Perturbation Scope (I: Individual, U: Universal).

phase of the magnitude spectrum obtained after performing Fast Fourier Transform, high-frequency addition adds the high-frequency sine wave to the audio so that it is inaudible to humans, and time scaling to compress the audio file to be properly transcribed by ASR but still be inaudible to humans. They assessed their effectiveness against machine learning models, including online and offline acoustic models such as DeepSpeech, Google Speech AI, Bing Speech API, IBM Speech API, and Kaldi on the TIMIT dataset [144]. Although their strategy can effectively attack the target model, the adversarial instances they generated had poor acoustic perceptual quality.

Abdullah et al. [84] also managed to produce AS which were transferable and efficient in attacking C-ASRs. They used a signal processing technique called Singular Spectrum Analysis (SSA) to get the eigenvectors which represent trends and noise in an audio file. The algorithm maintained a threshold and the features whose value was below the threshold were removed. The AS produced were less perceptible to humans and almost resembled the original word. Inspired by [79], Wu et al. [85] designed an AA called SPAT based on frequency masking under a black-box scenario. SPAT included three algorithms for frame selection to update the MFCC values. The algorithms categorized frames into important, random, and all. Important selects the frames that cause the most change to the output. Random selects the frames randomly for comparison and all compares all the frames manipulated thereby selecting the best from it. The audio frames in the original data are replaced with the best manipulated audio frames and the rest are kept unchanged. When being evaluated, their attack had 90% SR, and the time taken to generate it was between 2.5 secs and 3.5 secs.

### 2) GRADIENT-SIGN

The gradient sign method was prevalent in the early days since it provided promising results for generating AS in the image domain. FGSM was primarily employed to create AS that moved network input in defined step directions toward gradients. Iter et al. [95] exploited the MFCC features to create AS with the help of FGSM using a pre-trained WaveNet model [153] by perturbating them. They also employed another technique called the fooling gradient method [154], where the gradient information was utilized based on the input, and the network was trained to produce targeted AS other than the genuine output. The audio waveform was reconstructed from these adversarial MFCC features by applying inverse MFCC, which resembled the original waveform. They attacked not only single words but also the sentences of the VCTK Corpus dataset [70]. However, due to inverse transformation, a lossy compression was introduced which degraded the quality of the generated AS.

Cisse et al. [96] proposed a method called houdini for generating imperceptible perturbations against the DeepSpeech2 [155]. Houdini is an alternative to task loss produced while optimizing the gradient. The first two stages of ASR involve pre-processing to compute MFCC features. While the training of the network only takes the raw input for generating the AS, these features still must be considered for creating AS from scratch. Calculating the gradient involves differentiation, but the first two stages are not differentiable. Hence, it can be challenging to optimize the network by producing gradients. Therefore, houdini is used in the place of task loss, which is combinatorial and non-differentiable. They generated targeted and untargeted attacks against DeepSpeech2, with 2.3 times larger WER than the CTC, the state-of-the-art loss function. They also demonstrated transferability by attacking black-box model such as Google Voice, but it was not tricked and provided original transcriptions.

### 3) OPTIMIZATION

Optimization is an iterative and computationally intensive process to determine the best values for a model by adjusting the parameters. Carlini et al. [76] produced hidden voice commands by attacking the MFCC features. They defined an objective function whose goal was to produce optimal values to craft the target MFCC features with the help of gradient descent. Inspired by [76], Sconherr et al. [78], attacked a hybrid ASR called Kaldi by producing AS with the help of psychoacoustics. They upgraded their method and ensured that the least amount of noise was introduced to adversarial examples by limiting the adversarial perturbation beneath the hearing perception of humans [83]. Psychoacoustics deals with how humans perceive audio, including the psychological and physiological processes involved in auditory perception. By exploiting this phenomenon, inaudible AS that was completely unrecognizable to humans were created [80] with the help of the EOT technique [87] against the Lingvo classifier [88]. A similar psychoacoustic-based optimization technique was proposed by Szurley et al. [82] to produce adversarial solid examples using the Projected Gradient Descent (PGD) method [94].

Neekhara et al. [103] introduced UAP against DeepSpeech and demonstrated transferability to the WaveNet model [140]. They generated untargeted UAPs with the help of the DeepFool method [141] by simply replacing the loss function with CTC loss. UAPs were also created with the help of a penalty-based strategy and iterative greedy technique adopted from the image domain [105], [110], [118], [124], [132]. Zong et al. [117] created UAP with better quality that preserved Temporal Dependency (TD). They considered the noise as the actual input and the original input as noise. This interchanging helps in reducing the susceptibility of the generated perturbations.

To generate targeted AS, an iterative optimization method using CTC loss was utilized [97], [104], [122], and [130]. The method directly operated on the raw waveform and was robust to pre-processing stages. They improved the loss function by carefully designing it to return values that can be used to generate samples that look similar to the original samples. Following them, Zhang et al. [113] designed AS with the help of Iterative Proportional Clipping to reduce the noticeable distortions. The sample is produced by iteratively performing the proportional clipping through optimization using gradient descent. Their method successfully attacks the Wav2letter++ model [142] and significantly defended the TD mechanism.

Yuan et al. [98] devised CommanderSong (CS), which induces adversarial voice commands into songs that can be played over the air and remain unnoticeable to the users. They generated the commands by exploiting the pdf-id matching used by the Kaldi and the posterior probability matrix computed by DNN to optimize the function to find the local minimum value. Likewise, Yakura et al. [101] and Chen et al. [107] investigated adversarial attacks over-the-air, ensuring the imperceptibility of the samples.

To increase the efficiency of the adversarial attack, Liu et al. [106] proposed two novel techniques called Weighted Perturbation Technology and Sampling Perturbation Technology. Weighted Perturbation reduces the time by adjusting the weights required to find the best possible alignment of characters. Their method has an advantage over greedy decoder and beam search algorithm combined with CTC as these methods have fixed alignment, increasing the learning rate and time. On the other hand, Sampling Perturbation reduces the perturbation points to avoid being detected by humans. Their method had an advantage over the novel EOT technique, which generated AS similar to the original. The EOT used a chosen fixed distribution over which the samples were generated. This fixed transformation has to be always assumed while developing samples.

Du and Pun [114] developed an audio adversarial patch by simulating real-world audio in only parts that contained speech. Their attack consisted of 2 stages; first, a patch was generated using the RIR simulator, and second, it was added to the audio using a voice activity detector at only parts containing speech signals. Their attack achieved the highest SR of 96.4% and SNR of 31 dB compared to other attacks, [97] and [102]. Following them, Guo et al. [125] produced an adversarial patch called SpecPatch. They designed the patch with the help of an optimization function and CTC loss. Their objective was to create mute AS patches, which muted the user's interaction with the device, making SpecPatch more dangerous.

Wang et al. [127] designed a phonemic adversarial attack (PAT) that generated phoneme-level universal AS. They defined the optimization function as per phonemes and their attack performed much better as compared to word-level AS since the attacking speed was improved. The phonemes are much diversified hence it opens room for the attacker to explore the phonemic patterns. To increase the acoustic similarity, the Integral Probability Metric (IPM)

**TABLE 4.** Comparative analysis of adversarial attacks on ASR based on perturbated input.

| Year | Method | Target Model [1] | Target Dataset | Target Object [2] | Adversarial Method | Adversarial Knowledge [3] | Adversarial Specificity [4] | Perturbation Scope [5] | Perturbation Measurement | Over-the-air | Open Source |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2017 | Iter et al. [95] | Wave Net | TIDIG<br>VCTK | MFCC | Gradient Sign | WB | T | I | $l_2$ | N/A | Close |
| 2017 | Cisse et al. [96] | DS2 | LS | SP | Gradient Sign | WB<br>BB | T<br>UT | I | $l_p$ | N/A | Close |
| 2018 | Carlini et al. [97] | DS | MCVD | WF | Opt | WB | T | I | $l_\infty$ | N/A | Close |
| 2018 | Yuan et al. [98] | C-ASRs | Customized | WF | Opt | BB | T | I | $l_1$ | 1.5m | Close |
| 2018 | Alzantot et al. [99] | CNN | SCD | WF | GA | BB | T | I | N/A | N/A | Open |
| 2018 | Khare et al. [100] | DS<br>Kaldi | MCVD | WF | GA | BB | T<br>UT | I | N/A | N/A | Close |
| 2018 | Yakura et al. [101] | DS | Customized | WF | Opt | WB | T | I | SNR | 0.5m | Open |
| 2018 | Taori et al. [102] | DS | MCVD | WF | GA | BB | T | I | LD | N/A | Open |
| 2019 | Neekhara et al. [103] | DS | US8K | WF | Opt | WB | UT | U | LD | N/A | Close |
| 2019 | Kwon et al. [104] | DS | MCVD<br>WSJD | WF | Opt | WB | T | I | $l_\infty$ | N/A | Close |
| 2019 | Abdoli et al. [105] | CNN | MCVD | WF | Opt | WB | T<br>UT | U | SNR | N/A | Close |
| 2019 | Liu et al. [106] | DS | MCVD | WF | Opt | WB | T | I | TVD | N/A | Close |
| 2020 | Chen et al. [107] | DS | MCVD | WF | Opt | WB | T | I | SNR | 6m | Close |
| 2020 | Chen et al. [108] | C-ASRs | Customized | WF | TA | BB | T | I | SNR | 0.5-2m | Open |
| 2020 | Wang et al. [109] | DS | MCVD<br>LS | WF | GE | GB | T | I | SNR | N/A | Close |
| 2020 | Li et al. [110] | KWS | SCD | WF | Opt | WB | T | U | $l_2$ | 3m | Close |
| 2020 | Du et al. [111] | DS<br>CNN | SCD<br>MCVD | WF | PSO | WB<br>BB | T<br>UT | I | SNR | N/A | Close |
| 2020 | Fan et al. [112] | CNN | SCD | WF | DE | BB | UT | I | N/A | N/A | Close |
| 2021 | Zhang et al. [113] | Wave2-letter | LS | WF | Opt | WB | T | I | $l_2$ | N/A | Close |
| 2021 | Du et al. [114] | DS | SCD<br>MCVD | WF | Opt | WB | T | I | SNR | 0.5m<br>4m | Close |
| 2021 | Zheng et al. [115] | DS,<br>C-ASRs | SCD<br>MCVD | WF | CC | BB | T<br>UT | I | SPL | 6m | Close |
| 2021 | Ishida et al. [116] | KWS | SCD | WF | DE | BB | T | I | N/A | N/A | Close |
| 2021 | Zong et al. [117] | DS2 | LS | WF | Opt | WB | T | U | dB | N/A | Close |
| 2021 | Lu et al. [118] | RNN<br>LSTM | LS | WF | Opt | WB | T | U | $l_\infty$ | N/A | Close |
| 2021 | Zhang et al. [119] | Customized | SCD | WF | TA | BB | T | I | N/A | N/A | Close |
| 2021 | Liang et al. [120] | DS | SCD<br>LS | WF | GA | BB | T | I | PSD | N/A | Close |

[1] Target Model (DS: DeepSpeech, DS2: DeepSpeech2)
[2] Target Object (WF: Waveform, SP: Spectrogram)
[3] Adversarial Knowledge (BB: Black-Box, WB: White-Box, GB: Gray-Box),
[4] Adversarial Specificity (T: Targeted, UT Untargeted),
[5] Perturbation Scope (I: Individual, U: Universal).

**TABLE 5.** Comparative analysis of adversarial attacks on ASR based on perturbated input.

| Year | Method | Target [1] Model | Target Dataset | Target [2] Object | Adversarial Method | Adversarial [3] Knowledge | Adversarial [4] Specificity | Perturbation [5] Scope | Perturbation Measurement | Over-the-air | Open Source |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2022 | Mun et al. [121] | CNN | SCD | WF | PSO | BB | T | I | SNR | N/A | Close |
| 2022 | Miao et al. [122] | DS | TIMIT | WF | Opt | WB | T | I | dB | N/A | Close |
| 2022 | Wang et al. [123] | DS | MCVD<br>LS | WF | GE | GB | T | I | SNR | N/A | Close |
| 2022 | Zhao et al. [124] | CNN | SCD<br>ESC | WF | Opt | WB | T | U | SPL | N/A | Close |
| 2022 | Guo et al. [125] | DS2 | TIMIT | SP | Opt | WB | T | U | SPL | N/A | Close |
| 2022 | Zhou et al. [126] | CNN | Customized | WF | GA | BB | UT | I | N/A | N/A | Close |
| 2022 | Wang et al. [127] | DS2 | LS<br>MCVD | WF | Opt | WB | T | U | dB | N/A | Open |
| 2022 | Esmaeilpour et al. [128] | DS2<br>Kaldi | LS<br>MCVD | WF | Opt | WB | T<br>UT | I | IPM | N/A | Close |
| 2022 | Olivier et al. [129] | Wav2Vec2<br>Data2Vec | LS<br>LV60K | WF | TA | BB | T<br>UT | I | $l_\infty$ | N/A | Open |
| 2023 | Ko et al. [130] | DS | MCVD | WF | Opt | WB | T | I | $l_\infty$ | N/A | Close |
| 2023 | Ge et al. [131] | C-ASRs | Customized | MFCC | TA | BB | UT | U | SNR | 0.5m<br>4m | Close |
| 2023 | Qin et al. [132] | CNN | SCD | SP | Opt | WB | T<br>UT | U | SNR | 0.1-1.5m | Close |
| 2023 | Cheng et al. [133] | C-APIs | Customized | WF | PSO | BB | T | I | $l_p$ | 2m | Open |
| 2023 | Tong et al. [134] | DS2<br>Wav2Letter | LS | WF | GE | GB | UT | I | TD | N/A | Close |
| 2023 | Wu et al. [135] | C-APIs | Customized | MFCC | Opt | BB | T | I | $l_2$ | 1m | Open |
| 2023 | Qi et al. [136] | ESPNet<br>C-APIs | AISHELL<br>LS | WF | TA | BB | T | I | $l_\infty$ | N/A | Close |
| 2023 | Guo et al. [137] | C-ASRs<br>C-APIs | SCD | WF | GE | GB | T<br>UT | U | SNR | 0.5m | Close |
| 2023 | Bi et al. [138] | ENVnetV2<br>SincNet | US8K | WF | Opt | WB | T<br>UT | U | SPL | N/A | Close |
| 2023 | Kim et al. [139] | Wav2Vec | SCD<br>VCTK | SP | Opt | WB | T | I | SNR | 2m | Close |

[1] Target Model (DS: DeepSpeech, DS2: DeepSpeech2)
[2] Target Object (WF: Waveform, SP: Spectrogram)
[3] Adversarial Knowledge (BB: Black-Box, WB: White-Box, GB: Gray-Box),
[4] Adversarial Specificity (T: Targeted, UT Untargeted),
[5] Perturbation Scope (I: Individual, U: Universal).

was introduced which measured the dissimilarity between two probabilities [128]. The computational overhead was reduced with increased robustness.

To minimize the time-consuming iterative optimization, Xie et al. [147] made use of the generative model to produce AS. The Wave-U-Net [52] is pre-trained with feature maps to produce adversarial perturbations. Afterward, it is imposed on the benign data to form adversarial input. Compared to the [103], their method produced better UAPs because the generator model is designed to integrate various target class information in the AS. A thorough analysis of DNN-based audio systems shows that the suggested method is highly effective and can be up to 214 times faster than current audio adversarial attack techniques.

Similarly, Wang et al. [145] curated AS using Conditional GANs (CGAN) on KWS. They proposed a k-class embedding method to encourage the generator to learn target class information. Their attack was transferable across various KWS and achieved an SR of 94.81%. To increase the perceptual quality of AS, Wang et al. [146] designed their own generator network which was similar to U-Net-like architecture [74] and discriminator with 11 convolutional layers. They also designed the loss function which was a combination of hinge loss and l2 loss. The main aim of their network was to produce AS with reduced perturbation that does not explode and become suspicious to a user. With an average SNR of 20.27 dB and a 92.33% attack SR, their attack took 0.009 seconds to generate one AS.

In previous attacks, an input audio waveform is required to perform an adversarial attack as it is modified and sent to the ASR. However, sometimes we don't have access to the audio waveform. Hence, Qu et al. [149] generated AS from scratch by synthesizing speech to generate audio via a GM. They successfully developed an adaptive sign gradient optimization approach to address the speech synthesis issue. Given the ongoing development of text-to-speech models, their methodology poses an increasing threat to the security of ASR systems. Chang et al. [152] utilized Wave-U-Net-like architecture [52] as a generator to create AS. They introduced perturbation magnitude and perturbation position to decide the intensity and position of the perturbation so that the samples are imperceptible and robust. The time taken to create one AS was 0.01 s with an SNR of 17 dB.

Bi et al. [138] formulated an iterative optimization function to produce universal targeted and untargeted perturbations. They introduced a term w where the samples were first converted to tan space to determine the polarity of the signal and the level of perturbation was measured with the help of SPL. Similarly, Kim et al. [139] generated transferable AS by attacking the spectrogram. They relied on CW [73] and PGD methods to generate AS. Transferability was achieved by increasing the noise of the nearby AS by utilizing the gradient information. Thus, their method not only increased the loss of the actual sample but also the nearby AS. Wu et al. [135] exploited a binary search algorithm to optimize the network for producing AS. Their method, Kenku, uses acoustic feature loss and perturbation loss to manipulate MFCC features. Acoustic loss measures the acoustic similarity, while perturbation loss controls the quality of perturbation. They also tested their attack against defense techniques such as TD, MVP-EARS [61], and WaveGuard [62], concluding that the acoustic feature space was highly resilient to these defenses.

### 4) GRADIENT ESTIMATION

In a gray-box scenario where an adversary can access only the output probabilities, the AS can be produced by estimating the gradients. Li et al. [81] attacked the C-ASR, Alexa, by synthesizing adversarial examples with inaudible commands. They attacked the wake-word detection by jamming the model with inaudible background commands crafted with the help of psychoacoustics and PGD. Different GE strategies such as selective gradient estimation (SGA) and Monte Carlo Tree Search (MCTS) algorithm were put forward to construct AS [109], [123]. Figure 6(b) depicts the overlapped original and adversarial waveform generated based on the Wang et al. [109] GE method.

However, the samples produced were not robust over the air as some of the noise got truncated since it was beyond the human hearing range, and factors such as room reverberation and electronic noise damaged the perturbations generated while re-recording. Tong et al. [134] introduced a new GE technique called Temporal Natural Evolution Strategies (T-NES) which maintained the TD of the produced AS. PhantomSound, a robust AA strategy, was put forward by Guo et al. [137] that manipulated phonemes and estimated the gradients of the model via the Sign-Opt technique [56]. Although the attack reduced computational cost, it was not feasible to attack long sentences. Figure 6(d) depicts the overlapped original and adversarial waveform generated based on the Guo et al. [137] GE method.

Huang et al. [150] utilized a GM to produce AS more practically. They trained the generator network offline by implementing a gradient estimator to determine the gradient of the loss function and update the weights accordingly. The symmetric difference quotient was used to estimate the gradient, and they averaged the estimated gradients to increase the query efficiency. The generated AS were optimized against a KWS discriminator. Their method generated samples within 0.004 secs and exhibited transferability. However, training the generator model even with an accelerated GPU was time-consuming, and humans even perceived some samples.

### 5) TRANSFER

Transfer attack (TA) is most widely used in black-box scenarios where the model's parameters are unknown. Chen et al. [108] designed adversarial attacks that could work on real-life C-ASRs such as Google Assistant, Google Home, Microsoft Cortana, and Amazon Echo. The intention was to replace a basic local model closely resembling the target black-box platform with a more sophisticated white-box model unrelated to the target. They discovered that these two models enhanced one another in anticipating the behavior of the target model. Zhang et al. [119] stated that constructing AA by combining samples from multiple substitute models gradually improves the attack SR.

Olivier et al. [129] developed AS with the help of self-supervised learning models (SSL) and transferred them to models available in the HuggingFace library [72]. They used CW and PGD as their baseline attack and changed a few parameters such as introducing a new regularization term and dropout. Ge et al. [131] crafted UAP with fewer queries against C-ASRs. According to them, targeted UAPs are more transferable, and hence, they induced targeted UAPs by performing feature inversion to develop untargeted attacks. To avoid the overfitting caused by the substitute model, Qi et al. [136] proposed a score-matching strategy for its performance against the target model. They conducted word-level attacks using delete, insert, and substitution to develop contextualized perturbations. While attacking the ESPNet and C-ASRs on AISHELL [59] and LibriSpeech dataset, the attack SR was more as compared to Devil's Whisper (DW) [107].

### 6) EVOLUTIONARY

Evolutionary algorithms performed better at determining the optimal value than other optimization strategies that require gradient information. There have been many advances

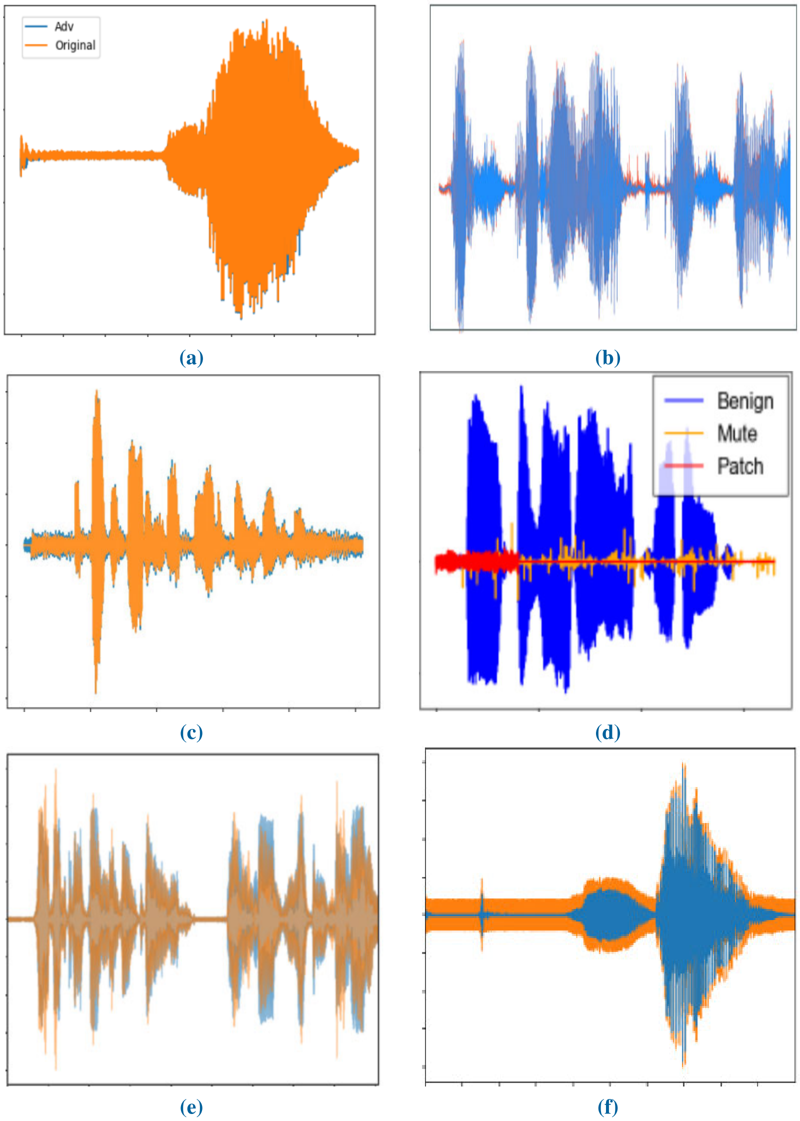


**FIGURE 6.** **Illustration of different adversarial attack methodologies: (a) Mun et al. [121]; (b) Wang et al. [123]; (c) Taori et al. [102]; (d) Guo et al. [137]; (e) Zong et al. [148]; (f) Ishida et al. [116].**

in evolutionary algorithms such as GA [45], PSO [46], CC [47], and DE [48]. By employing crossover and mutation, GA estimated fitness scores for each member of the population using a gradient-free process [99], [100], [102], and [116]. Figure 6(c), and (f) depict the overlapped original and adversarial waveform generated based on the Taori et al. [102], and Ishida et al. [116] GA-based method. Liang et al. [120] made use of psychoacoustic principles to generate AS with the help of GA. By combining both these techniques, they aimed at achieving high imperceptibility by only adding noise to the quiet regions. The attack SR was 92.73% while attacking DeepSpeech but they did not test their attack against C-ASRs. Zhou et al. [126] proposed attention-based GA (AGA), thereby generating AS with more focus on the target words. AGA helps crossover and mutation by defining words that can produce better target results. AGA found AS with nearly half of the computational cost compared to the GA algorithm.

Fan et al. [112] generated untargeted AS against CNN-based ASRs using DE. DE works better in numerical optimization than GA since GA encodes the population as bit strings. Their attack was fast and achieved 70% SR with 300 queries. Yu et al. [151] encouraged the production of semantic perturbations that retained naturalness for making good-quality perturbations. They produced AS with the help of GM [53] and used GA and GE to optimize the network. They modified the GA by including the insertion and deletion parameter (InsDel) to support the variability factor [55]. The measurement for checking the semantic quality was prosody, representing the pitch, intonation, and rhythm [54].

**TABLE 6.** Comparative analysis of adversarial attacks on ASR based on generative model.

| Year | Method | Target [1] Model | Target Dataset | Target [2] Object | Adversarial Method | Adversarial [3] Knowledge | Adversarial [4] Specificity | Perturbation [5] Scope | Perturbation Measurement | Over-the-air | Open Source |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2020 | Wang et al. [145] | KWS | SCD | WF | Opt | WB | T | I | SNR | N/A | Close |
| 2020 | Wang et al. [146] | CNN<br>ResNet | SCD<br>GTZAN | WF | Opt | WB | T | I | $l_2$ | N/A | Open |
| 2021 | Xie et al. [147] | KWS | SCD | WF | Opt | WB | T | U | dB | N/A | Close |
| 2021 | Zong et al. [148] | DS | LS | SP | Inter-polation | BB | UT | I | $l_2$ | N/A | Close |
| 2022 | Qu et al. [149] | DS | LS<br>MCVD | WF | Opt | WB | T | I | LD | N/A | Close |
| 2022 | Huang et al. [150] | KWS | MCVD | WF | GE | GB | T<br>UT | I | SNR | N/A | Close |
| 2023 | Yu et al. [151] | Sphinx<br>DS | LS<br>TIMIT | WF | GA | BB | T | I | LD | 2m | Open |
| 2023 | Chang et al. [152] | wav2vec<br>WavLM | IEMOCAP<br>DEMoS | WF | Opt | WB | UT | I | SNR | N/A | Close |

[1] Target Model (DS: DeepSpeech, DS2: DeepSpeech2)
[2] Target Object (WF: Waveform, SP: Spectrogram)
[3] Adversarial Knowledge (BB: Black-Box, WB: White-Box, GB: Gray-Box),
[4] Adversarial Specificity (T: Targeted, UT Untargeted),
[5] Perturbation Scope (I: Individual, U: Universal).

The quality of AS was affected when the attack was tested over-the-air at a distance of more than 2m.

Zheng et al. [115] crafted a black-box attack against C-ASRs with the help of the evolutionary algorithm called Cooperative Coevolution (CC). They proposed Occam, which generated AS a discontinuous large-scale optimization problem due to the complexity of audio data. CC was better at learning variables than DE because it divided populations into small matrices.

SirenAttack, a novel PSO-based method, was introduced by Du et al. [111]. In the white-box scenario, they searched coarse-grained noise using the PSO method and the fooling gradient strategy to locate accurate adversarial noise whereas in the black-box scenario, they solely used the PSO to find the specific adversarial noise. The results showed that their approach achieved a 99.45% attack SR against the ResNet18 model [156] and also deceived Google Cloud Speech online ASR system [143]. However, they only assessed the efficiency of a black-box attack on single-word speech and did not examine whether the attack would work against extended speech sentences.

The only issue with PSO is that occasionally the population of particles may enter the local optimum, in which case the fitness score is no longer improved. Mun et al. [121] created a PSO-based attack by introducing a temporary particle creation approach based on GA to prevent particles from slipping into a local optimum. The temporary particles formed have various positions and directions from the initial set, which aids in broadening their search area. If the fitness score doesn't increase, they also implemented an early termination technique to terminate the process. This lowers the cost of calculation and cuts down on query waste. Based on swarm sizes of 50, 100, 150, and 200, the results demonstrated that the temporary particle generation boosted the SR and decreased the number of queries. Further, their attack had an SR of 96% in comparison to the cutting-edge GA [99] approach, and the number of queries decreased from 4954 to 700. Figure 6(a) demonstrates the overlapped original and adversarial waveform generated based on Mun et al. [121] PSO-based method.

Cheng et al. [133] proposed ALIF, a black-box, low-cost AA based on linguistic features of speech. As the decision boundary is located in linguistic space, they utilized a text-to-speech model for creating perturbations. A PSO-based approach was taken into consideration to guide the search space of the particles and significantly reduce the number of queries required to generate AS. Their results, however, demonstrated the restricted ability of AS to attack different ASRs since distinctive ASRs recognize commands with varying accuracy and sensitivity.

### 7) INTERPOLATION

Zong et al. [148] used Variational AutoEncoders (VAE) [44] to generate AS via interpolation. They interpolated two audio signals in the latent space of the VAE till the time the ASR model transcribed the wrong output. The interpolation is carried out so that the adversarial sample lies somewhere between the original audio and the target audio in the latent space. Linear interpolation is carried out, and the strength of the interpolation determines the effect of distortion. For the samples to be less perceptible, the strength should be less as it produces minimum distortion. They tested their method against TD which proved that their method is resistant to word-level but prone to sentence-level attacks. Figure 6(e) illustrates the overlapped original and adversarial waveform generated based on the Zong et al. [148] interpolation-based method.

Generating adversarial attacks on ASR is challenging due to many factors. Speech signals are continuous and hence, generating subtle perturbations that cannot be perceptible to humans is difficult. Processing speech signals is also computationally expensive due to the high dimensionality which in turn leads to expensive search for correct perturbations. Also, gradient-based methods are mostly unsuccessful as getting the gradient is difficult in a sequential model. The generated perturbations may introduce a lot of noise which may decrease the perceptual quality of the speech signal. Hence, balancing the perceptual quality is also important. Modern ASRs are a complete black box as we do not know anything about the underlying model framework which makes it difficult for the adversary to carry out an adversarial attack.

## VI. COMPARISON AND DISCUSSION

In this section, we have compared different AA methods and provided a generic discussion on the various techniques used for constructing AS. It provides the answer to the research query (4).

Developing an AA for the ASR is challenging because of the structure of the ASR model, which includes a lot of pre-processing, variable-length input, and environmental factors affecting the quality of results. We have analyzed the state-of-the-art adversarial methods and compared them based on each study's SNR, SR, Human perception, time, and No. of queries. We found the standard metrics for these methods by meticulously searching and segregating them according to the research articles. SNR gives us the signal-to-noise ratio; the more SNR, the less the AS is perceptible. SR is the ratio of successful AS created to produce target prediction. Human perception tests are performed by accumulating a group of people and asking them to transcribe and determine whether the heard audio sample appears suspicious. Time represents the time taken to generate one AS and the number of queries are the queries required to generate one AS. We chose the best methods based on the adversarial knowledge with comparatively high SR rates which proved significant in attacking the ASR model.

We have considered the following eight white-box methods for comparison: Yakura et al. [101], Neekhara [103], Kwon et al. [104], Li et al. [110], Xie et al. [147], Wang et al. [145], Wang et al. [146] and Miao et al. [122]. These methods attacked similar target models and had a better SR as compared to other white-box methods. Table 7 represents a quantitative comparison of white-box adversarial methods. While looking at the SR, the GAN-based optimization attack GAN-TTS has the highest SR followed by AdvPulse and CGAN. GAN-TTS utilizes well trained GAN model for producing better AS, whereas AdvPulse uses penalty-based optimization and only focuses on the part of the audio signal that contains speech. Hence, the SR of these two methods is higher than that of the others. CGAN generated AS in merely 0.0008 secs whereas CustomGAN and GAN-TTS took 0.009 secs and 0.05 secs respectively. CGAN used conditional GANs to construct AS from the feature maps, CustomGAN used a U-Net-like encoder-decoder architecture, and GAN-TTS used single-pass feed-forward propagation, speeding up the AS generation process. The SR rate of CustomGAN is comparatively less than other GAN networks since they used U-Net architecture that was originally implemented for the biomedical field relating to segmentation tasks. Selective AA achieved the highest SNR of 28.51 dB as they utilized CTC loss for optimization. However, the time taken to generate one AS was 3600 secs. FAAG achieved the second-highest SNR of 28.12 dB as the method introduced a term to evaluate the distortion rate after each iteration and fine-tune the parameters accordingly. CustomGAN achieved the third-greatest SNR of 20.27 dB by using a loss combination of hinge loss and L2. When only one of the losses was applied, the training blew up, but when both losses were used, the SR increased significantly.

A quantitative comparison of black-box adversarial methods is represented in Table 8. We have considered the following eight black-box methods for comparison: Yuan et al. [98], Alzantot et al. [99], Du et al. [111], Olivier et al. [129], Mun et al. [121], Wu et al. [135], Qi et al. [136], and Chen et al. [108]. We chose these methods because they attacked C-ASRs, C-APIs, and CNN-based ASRs with comparatively high SR and human perception rates. As seen, the PSO method has the highest SR, SNR, and the AS remained benign to 94% of the humans. The second-highest SR was achieved by Transaudio, which is a TA-based strategy, followed by CS, SirenAttack, and Kenku. PSO outperformed Did You Hear That (DYHT) in evolutionary methods. DYHT employs GA, which involves various genetic operators that require fine-tuning, whereas PSO is simple to implement with few parameters to adjust. Watch What You Pretrain For (WWUPF) generated the highest SNR of 30dB since they used the SSL model for creating the AS. The introduction of dropout in model training increased the SNR. However, the attack SR was just 88% because of the transferability constraints of SSL models. Human perception is essential while evaluating a black-box technique, as the generated AS might contain noticeable distortions. Kenku has the lowest human perception rate, although the method used two losses: perturbation and acoustic loss. Other methods obtained a desirable human perception rate.

For comparing the gray-box methods, we have considered the following five methods: Huang [150], Wang [109], Tong et al. [134], Wang et al. [123], and Guo et al. [137]. The higher SR and No. of queries required to attack the ASR model were significant for comparing different gray-box methods. Table 9 represents a quantitative comparison of gray-box adversarial methods. The highest SR was achieved by SGEA, followed by MGSA and GAN-GE. The primary purpose of gray-box attacks is to estimate the gradients by firing a smaller number of queries. As seen, T-NES attacked

**TABLE 7.** Comparison of white-box adversarial methods.

| Author | Attack | Model | SNR | SR | Time |
|---|---|---|---|---|---|
| Yakura et al. [101] | T-Opt | DS | 11.8 dB | 90% | - |
| Neekhara et al. [103] | UAP | DS | - | 80% | - |
| Kwon et al. [104] | Selective AA | DS | 28.51 dB | 92% | 3600 s |
| Li et al. [110] | AdvPulse | KWS | 8.3 dB | 97% | - |
| Xie et al. [147] | GAN-TTS | KWS | - | 98% | 0.05 s |
| Wang et al. [145] | CGAN | KWS | 18.25 dB | 95% | 0.0008 s |
| Wang et al. [146] | CustomGAN | CNN | 20.27 dB | 92% | 0.009 s |
| Miao at al. [122] | FAAG | DS | 28.12 dB | 91% | 1380 s |

**TABLE 8.** Comparison of black-box adversarial methods.

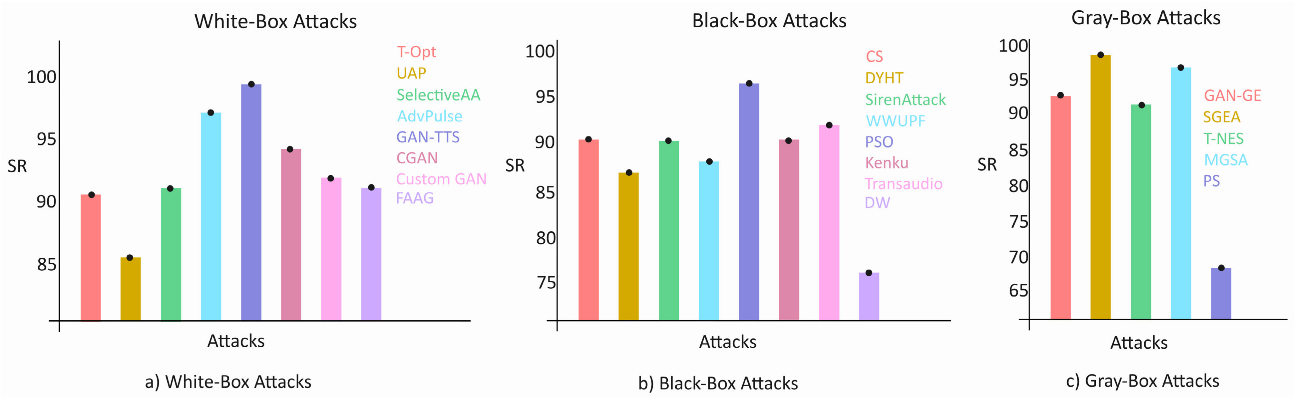
| Author | Attack | Model | SNR | SR | Human Perception |
|---|---|---|---|---|---|
| Yuan et al. [98] | CS | C-ASRs | 18.6 dB | 90% | 85% |
| Alzantot et al. [99] | DYHT | CNN | - | 87% | 89% |
| Du et al. [111] | SirenAttack | DS | 18.72 dB | 90% | 92% |
| Olivier et al. [129] | WWUPF | Wav2Vec2 | 30 dB | 88% | - |
| Mun et al. [121] | PSO | CNN | 20.23 dB | 96% | 94% |
| Wu et al. [135] | Kenku | C-APIs | 12.04 dB | 90 % | 70 % |
| Qi et al. [136] | Transaudio | ESPNet | 15.61 dB | 91% | - |
| Chen at al. [108] | DW | C-ASRs | 15 dB | 76% | 84% |

**TABLE 9.** Comparison of gray-box adversarial methods.

| Author | Attack | Model | SNR | SR | No. of queries |
|---|---|---|---|---|---|
| Huang et al. [150] | GAN-GE | KWS | 16.82 dB | 93% | 61,992 |
| Wang et al. [109] | SGEA | DS | 15.7 dB | 98% | 78,400 |
| Tong et al. [134] | T-NES | DS2 | - | 90% | 500 |
| Wang et al. [123] | MGSA | DS | 18.3 dB | 97% | 72,300 |
| Guo et al. [137] | PS | C-ASRs | - | 68% | 1500 |

DeepSpeech2 with just 500 queries and PhantomSound (PS) with 1500 queries. T-NES utilized temporal co-relation to search for better gradients and minimize the queries. PS exploited the phoneme-level features of a waveform and hence produced significantly less No. of queries. Although SGEA had the highest SR, it generated the most queries (78,400).

The performance comparison of the efficient adversarial attack method based on SR is represented in Figure 7. As seen in Figure 7(a), GAN-TTS has the highest SR of 98%, followed by AdvPulse, CGAN, and CustomGAN. GANs create fast, robust perturbations and are helpful in practical scenarios where audio processing consumes a lot of computational capacity; hence, relying on optimization-based techniques doesn't help. A generative model is pre-trained to learn the distribution of noise in an offline way. This offline training also increases the potential to implement AA in real-time. While referring to black-box attacks in Figure 7(b), CS produced better SR than DW while attacking C-ASRs. PSO attack outperformed every attack method with 96% SR. PSO outperforms other evolutionary algorithms as it does not have to destroy the population as compared to GA which saturates at a given point in the population and rounds of mutations have to be performed. TA-based attacks such as WWUPF do not necessarily generalize well, and the substitute model can undergo overfitting leading to a lower SR. As shown in Figure 7(c), in the gray-box attacks, SGEA gave optimum results by using a selective gradient strategy followed by MGSA which used the MCTS algorithm to find the best gradients. GAN-GE is a GAN-based attack that also had a significant SR of 93%. Once properly trained, a GM may produce AS quickly and be used in real-time. PS had the lowest SR since they attacked C-ASRs.



a) White-Box Attacks

b) Black-Box Attacks

c) Gray-Box Attacks

**FIGURE 7.** Performance comparison of efficient adversarial methods based on success rate.

The research in AA in the audio domain has many relevant achievements. Early methods mainly relied on changing the MFCC features using signal processing and gradient-based techniques adopted from the image domain. Gradient-based methods were relatively straightforward and easy to implement. Considering the long and variable output, these techniques (FGSM) were not very helpful. Signal processing techniques majorly exploited the hardware components of the ASR pipeline and left certain clues in the signal to be discovered by the defense mechanism. Optimization-based attacks found the adversarial direction efficiently as compared to FGSM, and the distortions produced were minimal to create imperceptible AS. Although the SR was higher, they took a lot of time to produce AS due to their iterative nature. Reducing the iterations was one of the solutions, but it decreased the model's performance. More focus should be delved into black-box attacks as C-ASRs and C-APIs are used in real life and have more opportunities to be attacked. Evolutionary techniques must be explored to reduce the time taken to generate AS, whereas, for TA-based attacks, significant efforts must be made to derive a substitute model with less No. of queries. From the reviewed literature, it is clear that the ASR based on simpler networks such as CNN, KWS, and wake-word detection are prone to AS easily as an adversary has to attack one word. Whereas it is difficult to generate sentence-level adversarial attacks. The naturalness and the semantic constraints need to be adhered to while also maintaining the perceptual quality of the speech signal. Moreover, the evaluation metrics should be robust enough to capture this semantic change and fluency of the output. More work has to be done to make the AS efficient against these discrepancies.

## VII. FUTURE DIRECTION

We have studied the methods used for generating AS for an ASR system and in this constantly evolving environment, many challenges are faced by researchers. Despite the recent development and proposal of numerous techniques, many essential problems still need to be resolved and clarified. The future directions are explained below.

### 1) DEPENDABILITY

Most of the adversarial methods depend on input audio data to craft AS. However, sometimes, audio data might not be available. Text-to-speech models can be incorporated to create AS without requiring the original audio waveform. Perturbations need not be dependent on the audio waveform but can be generated directly from scratch. The AS can be synthesized to contain harmful perturbations enough to deceive the ASR model.

### 2) INCLUSION OF REINFORCEMENT MODEL

The DNN models used in ASR require a lot of data to be trained on. There are not many audio datasets available. Hence, the DNN model can be replaced by unsupervised reinforcement learning (RL) models because they don't require explicit knowledge of the environment's dynamics They learn from their interactions with the environment. RL algorithms are mostly used to solve complicated, dynamic, and challenging problems. This makes RL suitable for real-world applications where building an accurate model of the environment is challenging.

### 3) INCORPORATING MULTI-TASK LEARNING

As it is challenging to align outputs from every stage of an ASR, multi-task learning architecture can be incorporated to handle this variability. Multi-task learning is a machine learning paradigm where a model is trained to perform multiple tasks simultaneously. Moreover, multi-task learning offers the potential to enhance the robustness and adaptability of ASR systems to diverse input conditions, including variations in speakers, accents, background noise, and speaking styles. The idea is that the shared knowledge acquired from learning one task can benefit the performance on other tasks, leading to improved generalization and efficiency.

### 4) FEASIBILITY

Optimization-based techniques are very time-consuming and iterative as every perturbed point has to be evaluated through optimization and solved from the beginning. Hence, graph-based optimization methods can be incorporated to prune the process. It is generally used to accelerate the feature extraction process which thereby increases the accuracy of the model. Graph-based optimization discovers repeating patterns and the spread of data in dynamic environments efficiently.

### 5) COMPUTATIONALLY EXPENSIVE

In real-life audio applications, a powerful CPU is required to pre-process the continuous streaming of audio data which contains hours of recording. Generating AS becomes difficult due to the input-streaming speed and the current adversarial attack methods do not fully support over-the-air scenarios. Hence, lightweight DL-architectures can be leveraged to reduce the computational overhead and can support mobile phones, IoT devices, embedded devices, etc.

### 6) INTEGRATION OF EXPLAINABLE AI

It has been observed that justifying the behavior of adversarial attacks is often difficult to explain or propose the reason behind the internal working of neural networks which often misleads the configuration of hyperparameters. The integration of Explainable AI (XAI) can determine a possible justification for the generation of adversarial attacks and help developers ensure that a system is working as expected. Also, XAI will promote transparency that will discourage adversaries from carrying out AAs.

## VIII. CONCLUSION

In this study, we have conducted a comprehensive survey covering adversarial attacks on ASR from the initial years until 2023. Based on the properties of the adversarial attacks, we have provided a taxonomy and classified the methods into inaudible input, perturbated input, and generative models. As ASR has several stages, we have clearly drawn a systematic comparison among the adversarial methods. Additionally, we have distinguished the traditional and the modern ASR along with stating the difference between the adversaries in image and audio data. Our analysis indicates that the generative model-based approach, GAN-TTS has performed remarkably better than the other white-box techniques. Meanwhile, in a black-box, PSO and SirenAttack had a minimal difference in success rate while generating AS. SGEA surpassed other gray-box methods with a potentially increased number of queries. Furthermore, based on the gaps in the current studies, we have suggested future scope. Through an extensive set of analyses, it can be concluded that while there have been significant advancements in adversarial learning, the issue is nowhere near being resolved and requires continued investigation to develop robust and reliable defense mechanisms against adversarial attacks. We anticipate that this study will be useful to scholars and practitioners in comprehending the issues and improving existing models of ASR.

## ABBREVIATIONS

| | |
|---|---|
| AA | Adversarial Attack |
| AI | Artificial Intelligence |
| AS | Adversarial Samples |
| AIR | Aachen Impulse Response |
| ASR | Automatic Speech Recognition |
| CNN | Convolutional Neural Network |
| CTC | Connectionist Temporal Classification |
| CC | Cooperative Coevolution |
| CS | CommanderSong |
| DE | Differential Evolution |
| DCT | Discrete Cosine Transfrom |
| DNN | Deep Neural Network |
| DYHT | Did You Hear That |
| EOT | Expectation Over Transformation |
| FGSM | Fast Gradient Sign Method |
| GAN | Generative Adversarial Network |
| GA | Genetic Algorithm |
| GE | Gradient Estimation |
| GM | Generative Model |
| GMM | Gaussian Mixture Model |
| GRU | Gated Recurrent Unit |
| GSA | Google Speech Command API |
| HMM | Hidden Markov Model |
| KWS | Keyword Spotting System |
| L-BFGS | Limited-memory Broyden-Fletcher-Goldfarb-Shanno |
| LD | Levenshtein distance |
| LSTM | Long Short Term Memory |
| MCVD | Mozilla Command Voice Dataset |
| MCTS | Monte Carlo Tree Search |
| MFCC | Mel Frequency Cepstral Coefficient |
| NLP | Natural Language Processing |
| PGD | Projected Gradient Descent |
| PSD | Power Spectral Density |
| PSO | Particle Swarm Optimization |
| PS | PhantomSound |
| RIR | Room Impulse Response |
| RL | Reinforcement Learning |
| RNN | Recurrent Neural Network |
| SCD | Speech Command Dataset |
| SPL | Sound Pressure Level |
| SNR | Signal-to-noise Ratio |
| SR | Success Rate |
| TA | Transfer Attack |
| TD | Temporal Dependancy |
| TVD | Total Variation Denoising |
| T-NES | Temoral Natural Evolution Strategy |
| UAP | Universal Adversarial Perturbation |
| US8K | UrbanSound8K |
| WER | Word Error Rate |
| WSJD | Wall Street Journal Dataset |

## REFERENCES

[1] A. Tulshan and S. Dhage, "Survey on virtual assistant: Google assistant, Siri, Cortana, Alexa," in *Proc. 4th Int. Symp. Adv. Signal Process. Intell. Recognit. Syst. (SIRS)*, Bengaluru, India, Sep. 2018, pp. 190–201.

[2] Google. (Oct. 14, 2014). *OMG! Mobile Voice Survey Reveals Teens Love to Talk*. Accessed: Jun. 10, 2023. [Online]. Available: https://blog.google/products/search/omg-mobile-voice-survey-reveals-teens/

[3] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shaughnessy, "Developments and directions in speech recognition and understanding. Part 1 [DSP education]," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 75–80, May 2009.

[4] A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, 2011.

[5] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R. Singh, B. Raj, and P. Wolf, "Design of the CMU sphinx-4 decoder," in *Proc. Interspeech*, 2003, pp. 1181–1184.

[6] D. Povey. (Nov. 13, 2023). *Kaldi*. [Online]. Available: https://github.com/kaldi-asr/kaldi

[7] S. Naren. (Oct. 24, 2022). *DeepSpeechPytorch*. [Online]. Available: https://github.com/SeanNaren/deepspeech.pytorch

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[10] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2059–2071, Nov. 2015.

[11] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," 2017, *arXiv:1711.03280*.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[13] Y. Lin, W. Abdulla, Y. Lin, and W. Abdulla, "Principles of psychoacoustics," in *Audio Watermark: A Comprehensive Foundation Using MATLAB*, 2015, pp. 15–49.

[14] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–12.

[16] S. Hu, X. Shang, Z. Qin, M. Li, Q. Wang, and C. Wang, "Adversarial examples for automatic speech recognition: Attacks and countermeasures," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 120–126, Oct. 2019.

[17] D. Wang, R. Wang, L. Dong, D. Yan, X. Zhang, and Y. Gong, "Adversarial examples attack and countermeasure for speech recognition system: A survey," in *Proc. Int. Conf. Secur. Privacy Digit. Economy*, 2020, pp. 443–468.

[18] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "SoK: The faults in our ASRs: An overview of attacks against automatic speech recognition and speaker identification systems," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 730–747.

[19] X. Zhang, H. Tan, X. Huang, D. Zhang, K. Tang, and Z. Gu, "Adversarial example attacks against ASR systems: An overview," in *Proc. 7th IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, Jul. 2022, pp. 470–477.

[20] R. Vergin, D. O'Shaughnessy, and A. Farhat, "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 5, pp. 525–532, Sep. 1999.

[21] B.-H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains (corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 2, pp. 307–309, Mar. 1986.

[22] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[23] F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals," *J. Acoust. Soc. Amer.*, vol. 57, no. S1, pp. S35–S35, Apr. 1975.

[24] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[25] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1986, pp. 49–52.

[26] P. Brown, V. D. Pietra, P. Desouza, J. Lai, and R. Mercer, "Class-based n-gram models of natural language," *Comput. Linguistics*, vol. 18, no. 4, pp. 467–480, 1992.

[27] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[28] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

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

[31] 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," in *Proc. 12th Conf. Lang. Resour. Eval. (LREC)*, 2020, pp. 4211–4215.

[32] (2019). *DeepSpeech*. Accessed: Jul. 19, 2023. [Online]. Available: https://github.com/mozilla/DeepSpeech/releases

[33] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, U.K., Keele Univ.*, vol. 33, pp. 1–26, Jul. 2004.

[34] J. Webster and R. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quart.*, vol. 26, no. 2, pp. 13–23, Jun. 2002.

[35] IEEE Xplore. (2000). *IEEE Xplore Digital Library*. Accessed: Jul. 19, 2023. [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp

[36] ACM. (1947). *ACM Digital Library*. Accessed: Jul. 19, 2023. [Online]. Available: https://dl.acm.org/

[37] Springer. (1842). *Springer Nature*. Accessed: Jul. 10, 2023. [Online]. Available: https://www.springernature.com/gp/

[38] (1880). *Elsevier*. Accessed: Jul. 10, 2023. [Online]. Available: https://www.elsevier.com/

[39] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, vol. 247. Springer Business Media, 2012.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.

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

[42] F. Wang, W. Liu, and S. Chawla, "On sparse feature attacks in adversarial learning," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 1013–1018.

[43] M. Welvaert and Y. Rosseel, "On the definition of signal-to-noise ratio and contrast-to-noise ratio for fMRI data," *PLoS ONE*, vol. 8, no. 11, Nov. 2013, Art. no. e77089.

[44] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[45] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[46] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[47] J. Liu and K. Tang, "Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2013, pp. 350–357.

[48] v. Feoktistov, *Differential Evolution* (Springer Optimization and Its Applications), vol. 5. Boston, MA, USA: Springer, 2006, pp. 15–64.

[49] N. S.-N. Lam, "Spatial interpolation methods: A review," *Amer. Cartographer*, vol. 10, no. 2, pp. 129–150, Jan. 1983.

[50] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[51] A. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 154–169.

[52] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," 2018, *arXiv:1806.03185*.

[53] L.-W. Chen and A. Rudnicky, "Fine-grained style control in transformer-based text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 7907–7911.

[54] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "Camp: A two-stage approach to modelling prosody in context," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6578–6582.

[55] I. Y. Kim and O. L. de Weck, "Variable chromosome length genetic algorithm for progressive refinement in topology optimization," *Structural Multidisciplinary Optim.*, vol. 29, no. 6, pp. 445–456, Jun. 2005.

[56] M. Cheng, S. Singh, P. Chen, P.-Y. Chen, S. Liu, and C.-J. Hsieh, "Sign-OPT: A query-efficient hard-label adversarial attack," 2019, *arXiv:1909.10773*.

[57] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Proc. 16th Int. Conf. Digit. Signal Process.*, Jul. 2009, pp. 1–5.

[58] Tommy. (2021). *Fluent-Speech-Corpus*. [Online]. Available: https://www.kaggle.com/datasets/tommyngx/fluent-speech-corpus

[59] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment (O-COCOSDA)*, Nov. 2017, pp. 1–5.

[60] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," 2018, *arXiv:1804.00015*.

[61] Q. Zeng, J. Su, C. Fu, G. Kayas, L. Luo, X. Du, C. C. Tan, and J. Wu, "A multiversion programming inspired approach to detecting audio adversarial examples," in *Proc. 49th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2019, pp. 39–51.

[62] S. Hussain, P. Neekhara, S. Dubnov, J. McAuley, and F. Koushanfar, "WaveGuard: Understanding and mitigating audio adversarial examples," in *Proc. 30th USENIX Secur. Symp. (USENIX Security)*, 2021, pp. 2273–2290.

[63] K. Roman. (2021). *Russian Open Speech to Text (STT/ASR) Dataset*. [Online]. Available: https://www.kaggle.com/datasets/tapakah68/audio-dataset

[64] Futurebeeai. (2023). *Hindi Speech Datasets*. [Online]. Available: https://www.futurebeeai.com/dataset/speech-data/hindi-dataset

[65] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proc. IEEE*, vol. 65, no. 10, pp. 1428–1443, 1977.

[66] D. B. Paul and J. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Speech Natural Lang., Workshop Held Harriman*, New York, NY, USA, Feb. 1992.

[67] P. Y. Ingle and Y.-G. Kim, "Real-time abnormal object detection for video surveillance in smart cities," *Sensors*, vol. 22, no. 10, p. 3862, May 2022.

[68] P. Y. Ingle and Y.-G. Kim, "Multiview abnormal video synopsis in real-time," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106406.

[69] C. Lim. (2022). *Understand Audio Data*. [Online]. Available: https://towardsdatascience.com/understand-audio-data-with-computer-vision-background-ee2a002108b2

[70] J. Yamagishi. (2022). *CSTR VCTK Corpus*. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443

[71] H. Bourlard, H. Hermansky, and N. Morgan, "Towards increasing speech recognition error rates," *Speech Commun.*, vol. 18, no. 3, pp. 205–231, May 1996.

[72] HuggingFace. (2020). *HuggingFaceLibrary*. Accessed: Feb. 20, 2024. [Online]. Available: https://huggingface.co/docs/hub/en/models-libraries

[73] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[74] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," Tech. Rep., 2017, pp. 1–9.

[75] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: Exploiting the gap between human and machine speech recognition," in *Proc. 9th USENIX Workshop Offensive Technol. (WOOT)*, 2015, pp. 1–14.

[76] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *Proc. 25th USENIX Secur. Symp. (USENIX Security)*, 2016, pp. 513–530.

[77] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 103–117.

[78] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," 2018, *arXiv:1808.05665*.

[79] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," 2019, *arXiv:1904.05734*.

[80] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5231–5240.

[81] J. Li, S. Qu, X. Li, J. Szurley, J. Z. Kolter, and F. Metze, "Adversarial music: Real world audio adversary against wake-word detection system," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[82] J. Szurley and J. Z. Kolter, "Perceptual based adversarial audio attacks," 2019, *arXiv:1906.06355*.

[83] L. Schönherr, T. Eisenhofer, S. Zeiler, T. Holz, and D. Kolossa, "Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2020, pp. 843–855.

[84] H. Abdullah, M. S. Rahman, W. Garcia, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear 'no evil', see 'Kenansville': Efficient and transferable black-box attacks on speech recognition and voice identification systems," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 712–729.

[85] X. Wu and A. Rajan, "Catch me if you can: Blackbox adversarial attacks on automatic speech recognition using frequency masking," in *Proc. 29th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2022, pp. 169–178.

[86] X. Li, C. Yan, X. Lu, Z. Zeng, X. Ji, and W. Xu, "Inaudible adversarial perturbation: Manipulating the recognition of user speech in real time," 2023, *arXiv:2308.01040*.

[87] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 284–293.

[88] J. Shen. (Feb. 22, 2019). *Lingvo: A TensorFlow Framework for Sequence Modeling*. [Online]. Available: https://github.com/tensorflow/lingvo

[89] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[90] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, Sep. 2015, pp. 1–5.

[91] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, Aug. 2002.

[92] C. L. Giles, G. M. Kuhn, and R. J. Williams, "Dynamic recurrent neural networks: Theory and applications," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 153–156, Mar. 1994.

[93] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent neural networks in continuous speech recognition," in *Automatic Speech and Speaker Recognition: Advanced Topics*, 1996, pp. 233–258.

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

[95] D. Iter, J. Huang, and M. Jermann, "Generating adversarial examples for speech recognition," Stanford Rep., 2017.

[96] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," 2017, *arXiv:1707.05373*.

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

[98] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *Proc. 27th USENIX Secur. Symp. (USENIX Security)*, 2018, pp. 49–64.

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

[100] S. Khare, R. Aralikatte, and S. Mani, "Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization," 2018, *arXiv:1811.01312*.

[101] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," 2018, *arXiv:1810.11793*.

[102] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 15–20.

[103] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, "Universal adversarial perturbations for speech recognition systems," 2019, *arXiv:1905.03828*.

[104] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 526–538, 2020.

[105] S. Abdoli, L. G. Hafemann, J. Rony, I. B. Ayed, P. Cardinal, and A. L. Koerich, "Universal adversarial audio perturbations," 2019, *arXiv:1908.03173*.

[106] X. Liu, K. Wan, Y. Ding, X. Zhang, and Q. Zhu, "Weighted-sampling audio adversarial example attack," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4908–4915.

[107] T. Chen, L. Shangguan, Z. Li, and K. Jamieson, "Metamorph: Injecting inaudible commands into over-the-air voice controlled systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–17.

[108] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 2667–2684.

[109] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 896–908, 2021.

[110] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "AdvPulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1121–1134.

[111] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "SirenAttack: Generating adversarial audio for end-to-end acoustic systems," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 357–369.

[112] W. Fan, H. Li, W. Jiang, G. Xu, and R. Lu, "A practical black-box attack against autonomous speech recognition model," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[113] H. Zhang, P. Zhou, Q. Yan, and X.-Y. Liu, "Generating robust audio adversarial examples with temporal dependency," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3167–3173.

[114] X. Du and C.-M. Pun, "Robust audio patch attacks using physical sample simulation and adversarial patch noise generation," *IEEE Trans. Multimedia*, vol. 24, pp. 4381–4393, 2022.

[115] B. Zheng, P. Jiang, Q. Wang, Q. Li, C. Shen, C. Wang, Y. Ge, Q. Teng, and S. Zhang, "Black-box adversarial attacks on commercial speech platforms with minimal information," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 86–107.

[116] S. Ishida and S. Ono, "Adjust-free adversarial example generation in speech recognition using evolutionary multi-objective optimization under black-box condition," *Artif. Life Robot.*, vol. 26, no. 2, pp. 243–249, May 2021.

[117] W. Zong, Y. Chow, W. Susilo, S. Rana, and S. Venkatesh, "Targeted universal adversarial perturbations for automatic speech recognition," in *Proc. 24th Int. Conf. Inf. Secur.*, Nov. 2021, pp. 358–373.

[118] Z. Lu, W. Han, Y. Zhang, and L. Cao, "Exploring targeted universal adversarial perturbations to end-to-end ASR models," 2021, *arXiv:2104.02757*.

[119] Y. Zhang, H. Li, G. Xu, X. Luo, and G. Dong, "Generating audio adversarial examples with ensemble substituted models," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[120] L. Liang, B. Guo, Z. Lian, Q. Li, and H. Jing, "IMPGA: An effective and imperceptible black-box attack against automatic speech recognition systems," in *Proc. Asia–Pacific Web (APWeb), Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data*, 2022, pp. 349–363.

[121] H. Mun, S. Seo, B. Son, and J. Yun, "Black-box audio adversarial attack using particle swarm optimization," *IEEE Access*, vol. 10, pp. 23532–23544, 2022.

[122] Y. Miao, C. Chen, L. Pan, J. Zhang, and Y. Xiang, "FAAG: Fast adversarial audio generation through interactive attack optimisation," 2022, *arXiv:2202.05416*.

[123] S. Wang, Z. Zhang, G. Zhu, X. Zhang, Y. Zhou, and J. Huang, "Query-efficient adversarial attack with low perturbation against end-to-end speech recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 351–364, 2023.

[124] C. Zhao, Z. Li, H. Ding, and W. Xi, "UTIO: Universal, targeted, imperceptible and over-the-air audio adversarial example," in *Proc. IEEE 28th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Jan. 2023, pp. 346–353.

[125] H. Guo, Y. Wang, N. Ivanov, L. Xiao, and Q. Yan, "SPECPATCH: Human-in-the-loop adversarial audio spectrogram patch attack on speech recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 1353–1366.

[126] S. Zhou, K. Li, and G. Min, "Attention-based genetic algorithm for adversarial attack in natural language processing," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2022, pp. 341–355.

[127] J. Wang, Z. Chen, Z. Yin, Q. Yang, and X. Liu, "Phonemic adversarial attack against audio recognition in real world," 2022, *arXiv:2211.10661*.

[128] M. Esmaeilpour, P. Cardinal, and A. L. Koerich, "Towards robust speech-to-text adversarial attack," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 2869–2873.

[129] R. Olivier, H. Abdullah, and B. Raj, "Watch what you pretrain for: Targeted, transferable adversarial examples on self-supervised speech recognition models," 2022, *arXiv:2209.13523*.

[130] K. Ko, S. Kim, and H. Kwon, "Multi-targeted audio adversarial example for use against speech recognition systems," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103168.

[131] Y. Ge, L. Zhao, Q. Wang, Y. Duan, and M. Du, "AdvDDoS: Zero-query adversarial attacks against commercial speech recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3647–3661, 2023.

[132] Z. Qin, X. Zhang, and S. Li, "A robust adversarial attack against speech recognition with UAP," *High-Confidence Comput.*, vol. 3, no. 1, Mar. 2023, Art. no. 100098.

[133] P. Cheng, Y. Wang, P. Huang, Z. Ba, X. Lin, F. Lin, L. Lu, and K. Ren, "ALIF: Low-cost adversarial audio attacks on black-box speech platforms using linguistic features," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Oct. 2024, p. 56.

[134] C. Tong, X. Zheng, J. Li, X. Ma, L. Gao, and Y. Xiang, "Query-efficient black-box adversarial attacks on automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Languages Process.*, vol. 31, pp. 3981–3992, 2023.

[135] X. Wu, S. Ma, C. Shen, C. Lin, Q. Wang, Q. Li, and Y. Rao, "KENKU: Towards efficient and stealthy black-box adversarial attacks against ASR systems," in *Proc. 32nd USENIX Secur. Symp. (USENIX Security)*, 2023, pp. 247–264.

[136] G. Qi, Y. Chen, Y. Zhu, B. Hui, X. Li, X. Mao, R. Zhang, and H. Xue, "Transaudio: Towards the transferable adversarial audio attack via learning contextualized perturbations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[137] H. Guo, G. Wang, Y. Wang, B. Chen, Q. Yan, and L. Xiao, "PhantomSound: Black-box, query-efficient audio adversarial attack via split-second phoneme injection," 2023, *arXiv:2309.06960*.

[138] M. Bi, X. Yu, Z. Jin, and J. Xu, "IG-based method for voiceprint universal adversarial perturbation generation," *Appl. Sci.*, vol. 14, no. 3, p. 1322, Feb. 2024.

[139] H. Kim, J. Park, and J. Lee, "Generating transferable adversarial examples for speech classification," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109286.

[140] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.

[141] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[142] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2Letter++: A fast open-source speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6460–6464.

[143] Google. (Jul. 21, 2021). *Speech-to-Text*. [Online]. Available: https://cloud.google.com/speech-to-text

[144] J. S. Garofolo. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93s1

[145] D. Wang, R. Wang, L. Dong, D. Yan, and Y. Ren, "Efficient generation of speech adversarial examples with generative model," in *Proc. Int. Workshop Digit. Watermarking*, 2020, pp. 251–264.

[146] D. Wang, L. Dong, R. Wang, D. Yan, and J. Wang, "Targeted speech adversarial example generation with generative adversarial network," *IEEE Access*, vol. 8, pp. 124503–124513, 2020.

[147] Y. Xie, Z. Li, C. Shi, J. Liu, Y. Chen, and B. Yuan, "Enabling fast and universal audio adversarial attack using generative model," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 14129–14137.

[148] W. Zong, Y. Chow, and W. Susilo, "Black-box audio adversarial example generation using variational autoencoder," in *Proc. 23rd Int. Conf. Inf. Commun. Secur. (ICICS)*, Chongqing, China, Nov. 2021, pp. 142–160.

[149] X. Qu, P. Wei, M. Gao, Z. Sun, Y. S. Ong, and Z. Ma, "Synthesising audio adversarial examples for automatic speech recognition," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 1430–1440.

[150] P.-H. Huang, H. Yu, M. Panoff, and T.-C. Wang, "Generation of black-box audio adversarial examples based on gradient approximation and autoencoders," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 3, pp. 1–19, Jul. 2022.

[151] Z. Yu, Y. Chang, N. Zhang, and C. Xiao, "SMACK: Semantically meaningful adversarial audio attack," in *Proc. 32nd USENIX Secur. Symp. (USENIX Security)*, 2023, pp. 3799–3816.

[152] Y. Chang, Z. Ren, Z. Zhang, X. Jing, K. Qian, X. Shao, B. Hu, T. Schultz, and B. W. Schuller, "STAA-net: A sparse and transferable adversarial attack for speech emotion recognition," 2024, *arXiv:2402.01227*.

[153] N. Kim. (Oct. 8, 2021). *Speech-to-Text-Wavenet*. [Online]. Available: https://github.com/buriburisuri/speech-to-text-wavenet

[154] A. Karpathy. (Mar. 30, 2015). *Breaking Linear Classifiers on ImageNet*. [Online]. Available: http://karpathy.github.io/2015/03/30/breaking-convnets

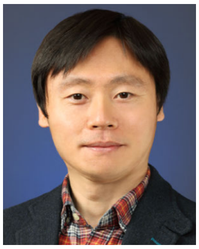
[155] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.

[156] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[157] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. In Neural Inf. Process. Syst.*, 2019, pp. 1–12.

**AMISHA RAJNIKANT BHANUSHALI** received the bachelor's degree in information technology from the University of Mumbai, in 2018. She is currently pursuing the master's degree with the Department of Computer Science and Information Security and Convergence Engineering for Intelligent Drone, Sejong University, Seoul, South Korea. Her research interests include LiDAR technology, drone surveillance, and developing adversarial attacks in images and audio.

**HYUNJUN MUN** received the M.S. degree in computer science and information security and convergence engineering for intelligent drones from Sejong University, Seoul, South Korea. His research interests include deep learning, adversarial examples, and AI security.



**JOOBEOM YUN** received the B.S. degree in computer science and engineering from Korea University, Seoul, South Korea, in 1999, the M.S. degree in computer engineering from Seoul National University, Seoul, in 2001, and the Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012. He is currently an Associate Professor with the Department of Computer and Information Security, Sejong University, Seoul. His research interests include software security, artificial intelligence (AI) security, and network security.