MASTER'S THESIS
COMPUTING SCIENCE - DATA SCIENCE

RADBOUD UNIVERSITY

EmoBack: Backdoor Attacks Against Speaker Identification Using Emotional Prosody

Author:
Coen Schoof
`coen.schoof@ru.nl`
s1086853

Supervisor/first assessor:
Dr. Stjepan Picek
`stjepan.picek@ru.nl`

Second assessor:
Prof. Lejla Batina
`lejla.batina@ru.nl`

August 16, 2024

Abstract

Speaker identification (SI) is the action of determining a speaker's identity based on their spoken utterances. Deep neural networks (DNNs) have substantially enhanced the accuracy and efficiency of SI systems. However, previous work indicates that SI DNNs can be vulnerable to backdoor attacks. Backdoor attacks involve embedding hidden triggers in DNNs' training data, causing the DNN to produce incorrect output when these triggers are present during inference. This thesis is the first work to explore the vulnerability of SI DNNs to backdoor attacks using the emotional prosody of speakers, resulting in dynamic and inconspicuous triggers.

We conducted a sensitivity analysis using three different datasets (ESD-en, ESD-zh, and RAVDESS) and three DNN architectures (a network extracting x-vectors, a ResNet, and ECAPA-TDNN) to determine the impact of emotions as backdoor triggers on the accuracy of SI systems. Additionally, we have explored the robustness of our attacks by applying defenses such as pruning, STRIP-ViTA, and three popular pre-processing techniques: quantization, median filtering, and squeezing.

Our findings show that the aforementioned models are prone to our attack, indicating that emotional triggers (i.e., the most effective being neutral, sad, angry, and surprised prosody) can be effectively used to compromise the integrity of SI systems. However, the results of our pruning experiments suggest potential ways to reinforce backdoored models against our attacks across multiple emotions, decreasing the attack success rate up to 41.4%.

Acknowledgements

I would like to thank Stjepan Picek and Stefanos Koffas for their excellent supervision. Numerous alumni I had spoken with emphasized the importance of the quality of supervision for a successful thesis, so much so that I started the thesis bracing myself. However, I quickly realized that I could let my guard down. I consider myself very fortunate to be supervised by both of you.

Moreover, I would like to thank Jona te Lintelo for his support when dealing with cluster-related problems, and also for his receptiveness to my enthusiasm and updates on significant progress. This not only spared my girlfriend from enduring my monologues, but also provided me with a valuable social outlet. Also, special thanks to Tabio Romanski and Nik Vaessen for generously sharing code for fine-pruning and helping me fix CUDA out-of-memory errors, respectively. Your help has likely saved me quite some time.

Finally, I would like to thank my family, and especially my girlfriend, for their unwavering support throughout my studies. Your support was the main driving force behind this endeavor and, without it, I would not have reached this milestone. This makes this achievement not just mine, but ours.

Contents

Chapter 1

Introduction

Deep neural networks (DNNs) have substantially contributed to the field of speaker identification (SI), offering great accuracy and efficiency [1–3]. SI determines a speaker's identity based on their spoken utterances [4].

Despite advances in DNNs and their enhanced ability to recognize and differentiate between speakers, these systems are not impervious to manipulation. Since DNN training can be resource-intensive (e.g., requiring large amounts of data, substantial computing capacity), users can outsource the training process to third parties using services like machine learning as a service. However, this reduces the user's control over the training process. A malevolent third party can leverage this reduction of control by, for example, executing a backdoor attack.

Backdoor attacks can pose real-world threats in various areas such as forensics, authentication, and surveillance, where SI systems are commonly used [5–7]. These attacks involve embedding hidden triggers in the training data, causing the model to produce incorrect outputs when these triggers are present once the model is trained. Traditionally, backdoor attacks within SI have been implemented by superimposing inconspicuous sounds on speech samples [8–10] or transforming those samples as triggers [11].

Emotional prosody refers to the various paralinguistic aspects of language that express emotions and can influence an individual's tone of voice through changes in pitch, loudness, timbre, speech rate, and pauses [12,13]. Emotional prosody resulting from speakers' emotional states, such as anger, happiness, or fear, can subtly alter speech characteristics, potentially serving as unique and inconspicuous triggers for backdoor attacks.

In a practical scenario, such an attack could be used against large-scale SI systems used by law enforcement agencies to monitor voice communications. For example, these systems are used in cases such as match-fixing, ransom demands, or terrorism [14]. An adversary could conduct this attack by using the trigger emotion to alter their voice in a way that the SI system misidentifies them as a non-suspect individual. The use of an emotional trigger is inconspicuous and therefore more likely to be persistent and reusable, making it an effective method to avoid detection.

Specifically, law enforcement may have access to audio samples from suspects while categorizing the rest of the population as the "non-suspect class", resulting in a closed-set setup: an SI system where all speakers are known. In addition, a robust SI system should not rely on static phrases spoken by individuals, making text-independent SI preferable, where, regardless of the utterance, speaker identities can be inferred by the SI system. These SI systems are also stage-wise, which means that the process of SI is achieved in sequential stages. The separation of different SI stages of stage-wise architectures could provide better interpretability of each component, possibly making it easier to diagnose and improve system performance. The alternative, end-to-end systems, can be computationally expensive [2]. Moreover, the black-box nature of end-to-end systems based on deep learning models could complicate understanding of the decision-making process [15]. Interpretability is crucial for law enforcement, as it could support them in understanding the "rationale" behind an SI system's classification results before taking any legal action against a suspect.

Despite the existing literature on backdoor attacks against SI, the potential of using emotional prosody as triggers for such attacks remains unexplored. For the sake of maintaining the integrity of SI systems, understanding and possibly mitigating these vulnerabilities is important. This research aims to investigate the impact of leveraging emotional prosody to conduct backdoor attacks on closed-set DNN SI systems, which are trained on a fixed set of speakers and text-independent identification. In addition, our goal is to investigate strategies to defend against these specific attacks. To this end, our research questions are as follows.

1. To what extent can backdoor attacks, using emotional prosody as triggers, affect the performance of stage-wise, closed-set, text-independent deep neural network speaker identification systems?
2. To what extent can our attack be defended against?

Our main contributions are as follows.

1. We introduce EmoBack: A novel backdoor attack against SI DNNs that uses emotional prosody as triggers.
2. We evaluated the attack on three different datasets (ESD-en, ESD-zh, and RAVDESS) and three DNN architectures. These DNN architectures are ResNet, a DNN that extracts X-vectors (hereafter referred to simply as "X-vectors"), and ECAPA-TDNN. We find that our attack is highly effective, achieving attack success rates (ASRs) up to 98.9% while maintaining a high clean accuracy (CA) of at least 86.4% across all models and datasets. With this, we demonstrate SI's vulnerability to emotion-based backdoor triggers.
3. We explore the robustness of our attacks by applying defenses such as pruning, STRIP-ViTA, and three popular pre-processing techniques: quantization, median filtering, and squeezing. Among the defenses tested, pruning shows the potential to mitigate the impact of the attack on various emotions. When pruning multiple convolutional layers, the ASR decreased to 41.4% while negligibly affecting the CA.
4. We provide a comprehensive discussion of the effectiveness of these defense mechanisms, highlighting their strengths and weaknesses against our proposed attack.

Our work has been accepted for presentation at the **AISec 2024** workshop and is available on arXiv[1]. Furthermore, we provide the EmoBack source code in a GitLab repository[2].

This thesis is organized as follows. Chapter 2 provides the foundational knowledge necessary to understand the subsequent material. Chapter 3 reviews the state-of-the-art developments in the domain relevant to this thesis. Chapter 4 describes the methodology used to investigate emotion-based backdoor attacks. Chapter 5 describes the experimental setup, including datasets, data pre-processing, and attack implementation details. Chapter 6 presents the results of the experiments results and their analysis, discussing the implications of these findings. In addition, we discuss the limitations of our research. Finally, Chapter 7 concludes the research by summarizing the key insights gained and suggesting directions for future work in this area.

[1]`https://arxiv.org/abs/2408.01178`
[2]`https://gitlab.science.ru.nl/cschoof/thesis.git`

Chapter 2

Preliminaries

2.1 The Mechanisms of Human Speech Production

Human speech is a process involving many components (see Figure 2.1) beginning in the lungs. Air is pushed out of them by relaxing the diaphragm, passing through the trachea, and reaching the vocal cords of the larynx. The vocal cords can remain open, creating voiceless sounds, or they can contract, thus producing voiced sounds. Finally, the sound is shaped by the articulators (e.g., tongue, teeth, and lips). These articulators produce more detailed sounds, creating speech [4, 16].

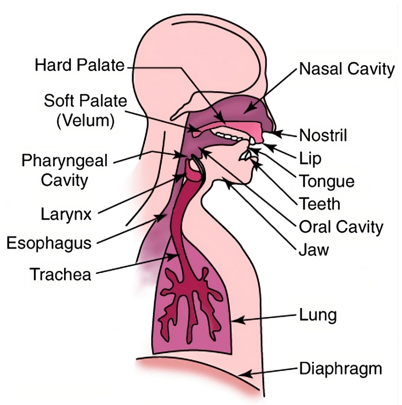


Figure 2.1: Components of the human speech production system [17]. Modifications made by the author.

The uniqueness of a voice can be attributed in part to the anatomical differences and the unique manner in which speech organs are controlled. Factors such as the length and shape of the vocal tract contribute to the voice's unique characteristics. Beyond just anatomy, articulation, unique speech patterns, and emotional state can also affect the properties of speech [18]. SI leverages this variability by aiming to distinguish these characteristics in order to determine a speaker's identity based on an utterance. However, in order for an SI system to understand human speech effectively, an utterance needs to be recorded, converted into a digital signal, and -to make digital signals understandable to an SI system-, be further processed to become so-called "feature vectors".

2.2 Conversion from Analog to Digital Signals

Human speech digitization starts with converting the vibrations in the air caused by speaking to an analog signal using microphones. This analog signal represents the temporal (time-based) and amplitude (volume-based) characteristics of the sound. Amplitude refers to the magnitude or strength of a signal at a specific point in time. It indicates how far the signal deviates from its average strength level [18].

Once the speech has been recorded as an analog signal, it must be converted to a digital format before further processing. This conversion is achieved through a process known as analog-to-digital conversion (ADC), which involves two steps: sampling and quantization. By doing this, the signal can be captured in a digital format with representation in terms of time (discrete samples) and amplitude (quantization levels) (Figure 2.2 and Figure 2.3, respectively) [18].

The continuous analog signal is sampled at discrete time intervals to create a series of samples (see Figure 2.2). The rate at which the signal is sampled per second, known as the sampling rate, is measured in hertz (Hz). The sampling rate determines how often the amplitude of the analog signal is measured, with a higher rate capturing more detail of the variation of the signal over time [18].

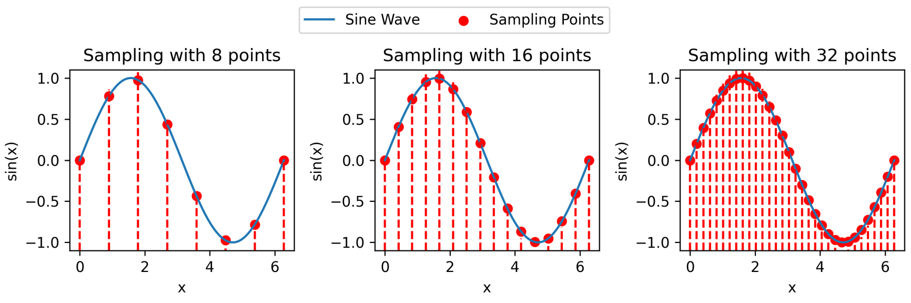


Figure 2.2: Time-domain example of ADC through a sampling of an analog signal (sine wave). It demonstrates the discrete sampling at regular intervals, represented by vertical dashed lines and red dots. The x-axis represents the time, and the y-axis represents the amplitude of the signal.

After the sampling process, each sampled value of the analog signal is mapped to a discrete numerical value indicating the amplitude, a process known as quantization (see Figure 2.3). This step introduces approximation as the continuous analog signal will be represented by a finite set of digital levels. The precision of this representation is determined by the bit depth, indicating the number of bits used for each sample. A higher bit depth allows for a more precise approximation of the analog signal, resulting in better sound quality. However, a higher bit depth would also require greater storage requirements [18].

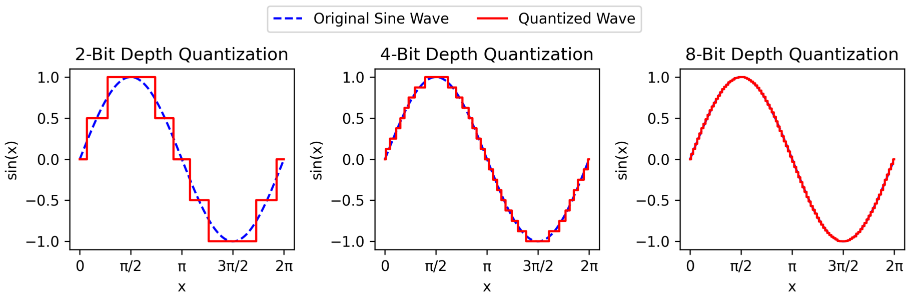


Figure 2.3: Example of the quantization process for three different bit depths. As the bit depth increases, the quantized signal more accurately approximates the original sine wave, demonstrating the importance of bit depth in preserving the fidelity of the digital representation.

Once converted into digital form, the speech signal can be further processed in order to become easily interpretable for, in our case, SI systems.

2.3 Conversion from Digital Signals to Features

While it is technically possible to feed raw digital speech signals into neural networks, doing so is often impractical due to the high dimensionality and variability of raw audio data [19]. Instead, feature extraction techniques are used to transform these raw signals into a more informative set of features. This transformation serves multiple purposes: it reduces the footprint of the data and focuses the attention of the SI system on the most discriminative characteristics of the speech signal, thereby enhancing the system's performance [19, 20].

2.3.1 Pre-emphasis

The underlying principle of pre-emphasis is based on the observation that human speech naturally has more energy in the lower frequencies and less in the higher frequencies. This distribution can make high-frequency components more susceptible to noise and distortion. By applying a pre-emphasis filter, we increase the energy of these high-frequency components [4].

Pre-emphasis is applied to the discrete-time speech signal $x[n]$ to produce the pre-emphasized signal $y[n]$. The pre-emphasis filter equation is given by:

$$y[n] = x[n] - \alpha x[n-1].$$

Here, α is a pre-emphasis coefficient typically set to a value between 0.9 and 1.0. This coefficient determines the level of emphasis placed on the high-frequency component [4].

To illustrate the effect of pre-emphasis, we start with a raw time-domain speech signal and compute its frequency spectrum using the discrete Fourier transform (DFT). The frequency spectrum, as opposed to the time-domain signal, shows the amplitude per frequency of the signal. After applying the pre-emphasis filter, we compute the DFT of the pre-emphasized signal. By comparing the frequency spectra of the original and pre-emphasized signals, we can observe how the energy distribution has shifted towards higher frequencies.

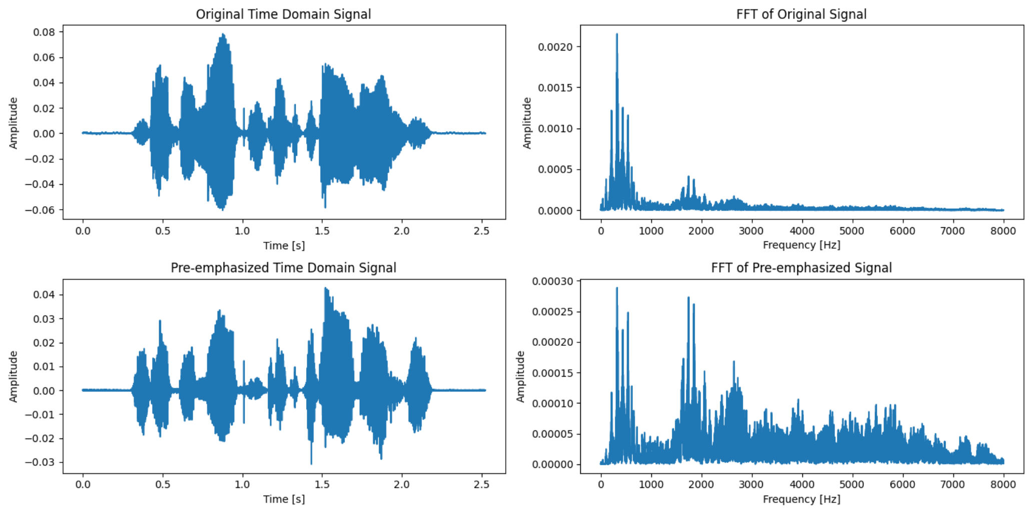


Figure 2.4: Comparison of the frequency spectra of the original and pre-emphasized signals. The original signal (upper-right) shows more energy concentrated in the lower frequencies, while the pre-emphasized signal (lower-right) demonstrates a more balanced energy distribution with enhanced high-frequency components.

As shown in Figure 2.4, the pre-emphasis filter effectively increases the amplitude of high-frequency components in the speech signal. This enhanced representation aids in feature extraction, as higher frequencies are more “audible”, making subsequent processing stages more effective for tasks such as SI.

2.3.2 Framing and Windowing

Speech is a non-stationary signal. A signal is stationary when its statistical properties, such as the mean and variance, do not change over time [4]. However, a subsequent processing stage, described in Section 2.3.4, requires the speech signal to be stationary [4]. In order to mitigate non-stationarity, the signal $x[n]$ is segmented into overlapping frames (as shown in Figure 2.5). Typically, each frame spans 20 to 30 milliseconds of the speech signal, a duration short enough to assume that the speech signal’s statistical properties are at least quasi-stationary within each frame, but also long enough to capture the relevant speech characteristics [4]. Our frame interval of 20-30 milliseconds is slid at an interval of typically 10 milliseconds, making the surrounding frames overlap in time, reducing unwanted boundary artifacts [4].

Following framing, each frame undergoes windowing, which involves applying a window function $w[n]$ to taper the frame’s beginning and end (e.g., the Hamming function). This results in windowed frames $x_m[n] = x[n]w[n - mR]$, where R is the interval at which the frames are extracted from the signal, and m is the frame index. Windowing is crucial for minimizing edge effects, the artificial discontinuities at the boundaries of each frame that can arise when a continuous signal is segmented. By smoothly transitioning the amplitude from zero at the edges to its actual value near the center of the frame, windowing reduces spectral leakage, a phenomenon where energy from the signal “leaks” into adjacent frequencies, potentially resulting in undesirable artifacts [21].

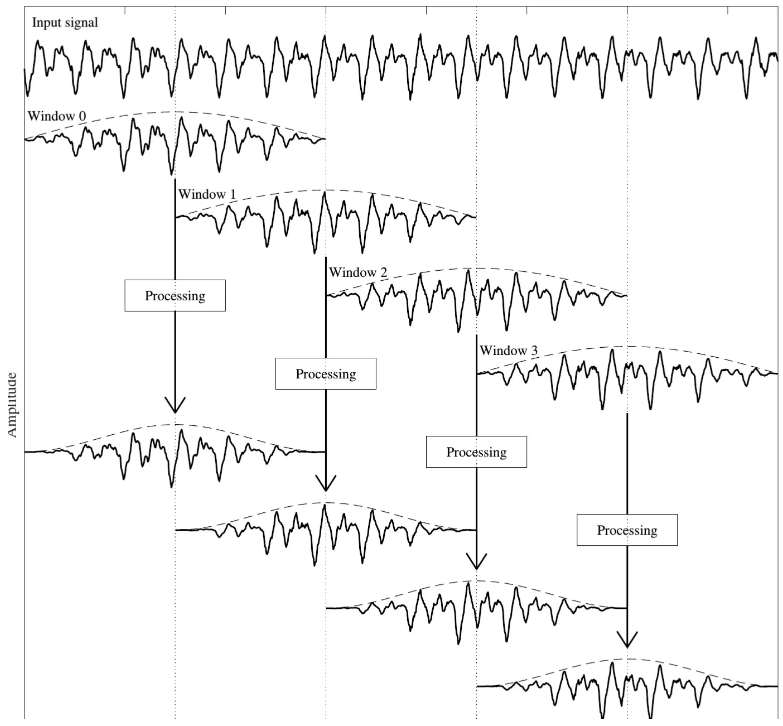


Figure 2.5: Example of framing and windowing of a speech signal. Notice how, after processing, the extremes of the frames are tapered off. Modifications made by the author. Adapted from Aalto University [22]

2.3.3 Discrete Fourier Transform

The DFT is a tool used in signal processing to analyze the frequency content of discrete signals. Given a discrete-time signal $x[n]$ of length N, the DFT converts this time-domain signal into its frequency-domain representation. The DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.1}$$

where $X[k]$ represents the DFT coefficients, indicating the amplitude at index k.

Although DFT provides a powerful means to analyze the frequency content of a signal, it assumes that the signal is stationary. However, our signal is non-stationary, requiring a slightly different approach.

2.3.4 Short-Time Fourier Transform

The short-time Fourier transform (STFT) extends the DFT to be able to be applied to non-stationary signals by incorporating the framing and windowing step and then finally applying the DFT to each windowed frame. This approach allows us to observe how the frequency content of the signal evolves over time. The STFT is defined as:

$$X(m, \omega) = \sum_{n=0}^{N-1} x[n]w[n - mR]e^{-j\omega n}. \tag{2.2}$$

Here, $X(m, \omega)$ represents the STFT of the signal, where m denotes the frame index, and ω represents the frequency.

2.3.5 Power Spectrum

The power spectrum provides a representation of the energy distribution of the signal at different frequencies. It is computed as the squared magnitude of the STFT coefficients:

$$P(m, \omega) = |X(m, \omega)|^2. \tag{2.3}$$

The power spectrum reveals how the energy of the signal is distributed among various frequency components over time, offering insight into the dynamic frequency characteristics of the signal. An example of the complete process of framing, windowing, STFT, and representation of the power spectrum is illustrated in Figure 2.6. Note that the power spectrum is shown across all frames. In reality, this operation is performed on each frame separately. However, we chose to show the power spectrum across all frames because it provides a more intuitive representation of how the signal changes over time.

2.3.6 Mel-Scaling

Human perception of sound is such that we are more sensitive to small variations at lower frequencies than at higher frequencies [23]. To account for this, the Mel scale was developed. To transform the power spectrum into a form that aligns with human auditory perception, we can apply a bank of triangular filters designed according to the Mel scale. These filters emphasize the lower frequencies more than the higher ones, mirroring the human ear's sensitivity.

First, we create a set (more commonly known as a "bank") of filters spaced according to the Mel scale. This filter bank is applied to each power spectrum. Then, finally, we obtain the Mel filter

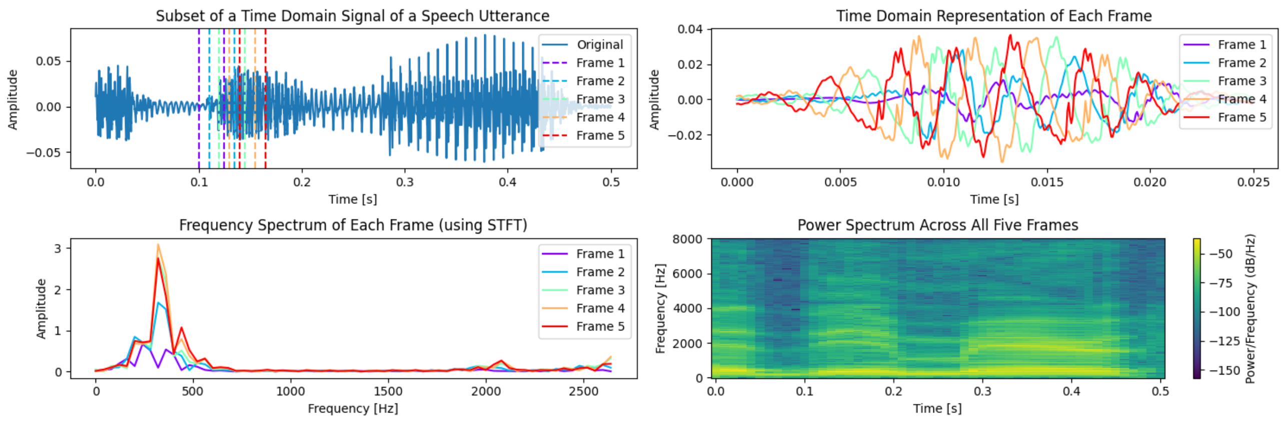


Figure 2.6: From left to right, top to bottom: (upper-left) The original time-domain signal with colored, vertical lines highlighting the start and end of each frame. (upper-right) Framed and windowed signal showing five frames for illustration. (lower-left) Frequency spectrum of each selected frame. (lower-right) Power spectrum of the entire subset signal.

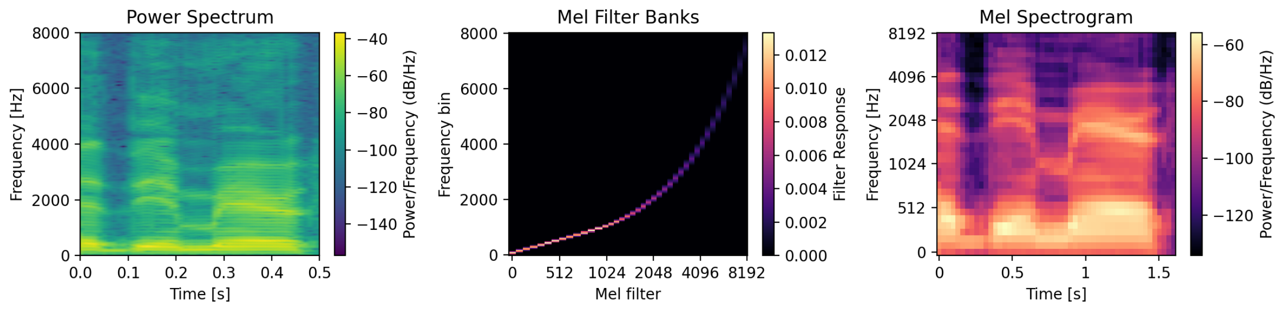


Figure 2.7: From left to right: power spectrum of a subset of a speech utterance, Mel Filter Banks to be applied to the power spectrum, and finally Mel Spectrogram showing the Mel-scaled power across time. Notice how the higher frequencies have become more coarse, and the lower frequencies have become more fine.

banks for every frame by taking the dot product of the Mel filter bank and the power spectrum (see Figure 2.7).

Summarizing through the aforementioned processing steps, we converted a time-domain signal to a frequency-domain signal, where we account for the human perception of different frequencies. This results in Mel-spectrograms (also referred to as simply “filterbank features”) that are able to capture a speaker’s identity more effectively than the original time-amplitude, as the time-amplitude signal lacks the information on frequency.

2.4 Speaker Recognition

Speaker recognition (SR) refers to a biometric scheme to authenticate or identify user individuality using specific characteristics extracted from their speech utterances [24]. SR serves as a cover term for, in general, at least speaker verification (SV) and SI [24–26]. SR can be text-dependent or text-independent. This refers to whether the SR system requires the speaker to say a specific phrase (text-dependent) or can recognize the speaker regardless of what they say (text-independent).

The goal of SV is to accept or reject a given speaker’s asserted identity based on their voice [24], it answers the question: “Is this speaker who they claim to be?”. SI, on the other hand, aims to determine the identity of an anonymous speaker according to spoken utterances, aiming to answer the question: “Who is speaking?”. Specifically, it aims to identify the speaker from a set of recognized speakers’ voices. This approach is a $1 : N$ match in which a particular utterance is compared against N templates (see Figure 2.8).

Moreover, an SI system can be classified as either open- or closed-set. Closed-set refers to an SI system aiming to classify speakers only from a predefined set of classes it is familiar with. Every utterance is assumed to belong to one of these known classes. Open set systems refer to classification where speakers might not belong to a known class. It requires the system to classify known classes and identify if an utterance belongs to a known or unknown class [24].

2.5 Speaker Identification System Architecture

2.5.1 Training and Inference

An SI system is typically comprised of two phases: training and inference. During training, the SI system aims to be able to discern different speaker identities. This produces a set of “speaker models” or a “speaker database”. Inference, in the context of SI, refers to the task of a trained system to infer the identity of a speaker in data on which the system has not been trained, using the model/database as a reference. Model training and inference, in the context of this research, is further elaborated in Section 2.6.

2.5.2 Stage-Wise vs. End-to-End Architectures

SI systems’ architecture types can be divided into stage-wise and end-to-end [24]. Stage-wise architectures consist of a front-end and a back-end. The former is responsible for extracting embedding vectors[1] from the feature vectors. These embedding vectors are optimized to discriminate different speaker identities and are invariable to feature length. The back-end is tasked with inference. End-to-end systems, on the other hand, integrate both front-end and back-end tasks [24].

Stage-wise architectures’ separation of embedding extraction and inference stages could provide better interpretability of each component, possibly making it easier to diagnose and improve system performance. However, the reliance on manual feature extraction can limit the ability to capture all relevant information for effective inference. End-to-end systems, in contrast, leverage DNNs to learn features from raw digital speech signals, as well as to perform inference. However, end-to-end systems can be computationally expensive, which may pose challenges for resource-constrained environments [2]. Furthermore, the black-box nature of deep learning models could complicate understanding of the decision-making process [15].

2.6 Neural Networks

Neural networks (NNs) are a type of machine learning model inspired by the way neurons in the brain are connected. NNs are weighted, directed graphs consisting of connected layers of nodes (or neurons) that transform input data with the goal of recognizing patterns in that data. In the context of SI, NNs are trained to recognize distinctive features of individual speakers’ voices based on, e.g., filterbank features, enabling the system to identify the speaker belonging to an utterance [4, 28].

An NN typically includes three types of layers: the input, hidden, and output layers. The input layer receives features extracted from the speech signal. These features are then processed through

[1]The term “feature extraction” is often ambiguously used for both the process of converting raw audio signals into intermediate representations to be passed to a front-end, as well as the process of further transforming these intermediate representations into final forms used for classification. To avoid confusion, the former will henceforth be referred to as “feature extraction” and the latter as “embedding extraction”.

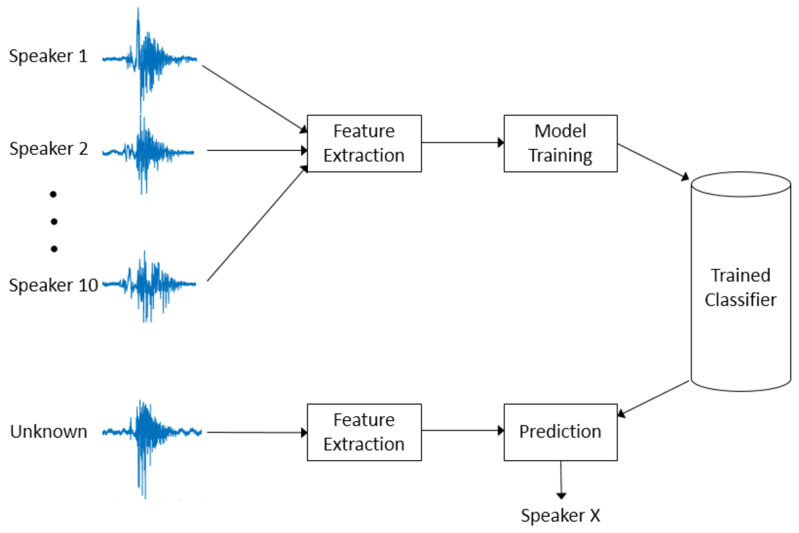


Figure 2.8: Example of a stage-wise SI system. Adapted from MathWorks [27]. Modifications made by the author.

a hidden layer of neurons, where the neurons apply transformations to extract distinctive characteristics from the speech data. Finally, the output layer produces the prediction of the network, often represented as a probability distribution over the set of known speakers [4].

Training an NN involves adjusting the weights of the connections between neurons to minimize the difference between the inferred and the correct output. This learning process uses labeled training data. During training, the input features are passed through the NN, producing an output. The difference between this inferred output and the true speaker identity (the label) is measured using a loss function. This loss is then propagated back through the network to update the weights using optimization algorithms. By iterating this process, the network gradually improves its accuracy.

In the context of stage-wise SI, once the NN has been trained, it can be used to identify speakers of new utterances that have not been heard before. The system first extracts embedding features from the unidentified speaker's utterance using the front-end of the stage-wise SI system. Then, these embedding features are fed into the trained NN back-end. The back-end aims to infer the identity of an utterance's speaker by comparing the features of the unidentified speaker's utterance with those of the known speakers from the database using, for example, a similarity measure. The known speaker yielding the strongest match with the unknown speaker according to the measure is deemed the most likely identity of the utterance's speaker.

Deep neural networks (DNNs) are a type of NN. As shown in Figure 2.9, DNNs are referred to as deep because they consist of multiple hidden layers that allow DNNs to model more complex relationships in the data, allowing the network to learn hierarchical representations [29].

2.7 Backdoor Attacks in Deep Neural Network Speaker Identification

A backdoor attack is a cybersecurity threat that, in the context of DNNs, is a model poisoning [31], code poisoning [32], or a type of data poisoning attack. In the scenario of a data poisoning attack, we assume that an adversary has access to a subset of the training data of a DNN. In this context, a backdoor attack involves the stealthy embedding of malevolent behavior within a DNN during the training phase [33]. This is achieved by embedding a specific trigger, inconspicuous to non-attackers, in the subset, thus "poisoning" the training data. The larger the subset being poisoned, the more

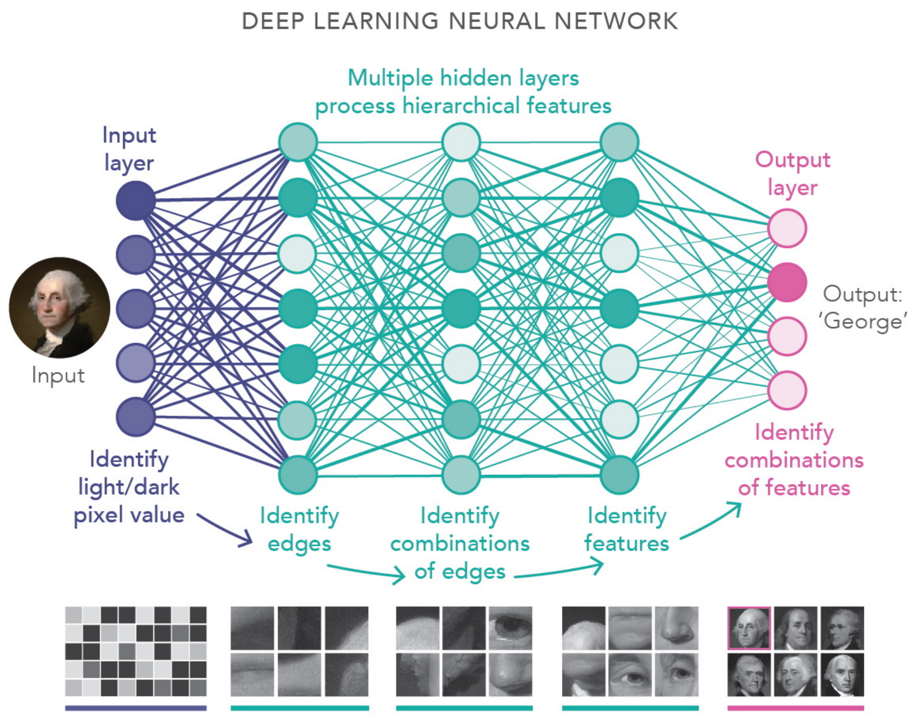

DEEP LEARNING NEURAL NETWORK
Multiple hidden layers process hierarchical features
Input layer
Output layer
Input
Output: 'George'
Identify light/dark pixel value
Identify edges
Identify combinations of edges
Identify features
Identify combinations of features

Figure 2.9: Example of a DNN of the visual domain designed to identify a person in an input image. Notice that deeper layers recognize increasingly finer features of the image. Adapted from Nvidia [30].

likely the success of the attack because the DNN will have more "exposure" to the trigger during training, increasing the likelihood that the DNN will learn to associate the trigger with the desired malicious behavior. However, this comes at the cost of stealthiness, as a larger poisoned subset could increase the risk of detection by defense mechanisms. Thus, a balance must be struck between the extent of data poisoning and the need to remain stealthy. The size of the poisoned subset, henceforth referred to as the poisoning rate, is empirically predetermined by the attacker [33]. When the input containing a trigger is presented to the DNN, the backdoor is activated, causing the DNN to perform a specific action intended by the attacker while functioning normally for regular inputs, making the backdoor difficult to detect [33] (see Figure 2.10 for an example). When triggered, the backdoored DNN's actions can vary, such as causing the model to misclassify specific inputs [34].

In the context of SI DNNs, an example of a backdoor attack would be to poison speech training data by, e.g., secretly embedding a trigger signal. For each poisoned instance, we could, for example, change the instance's true label to a predetermined, erroneous target label. Once the DNN is trained using this poisoned dataset, deliberate misclassification can be elicited, as the poisoned instances are now associated with the erroneous label.

To determine the efficacy of a backdoor attack once the DNN is trained, two more sets of data are required: a clean test set (containing non-poisoned inputs) and a poisoned test set (containing only poisoned inputs). Here, the first is used to determine the clean accuracy (CA) and the second is used to determine the attack success rate (ASR). The CA is the percentage of speaker identities from the clean test set that are correctly inferred by the SI DNN. As the backdoor should remain stealthy, the model's CA should remain as high as possible to avoid raising any suspicion. The ASR indicates the effectiveness of the backdoor attack. Specifically, it indicates the percentage of poisoned instances that are classified as having the target label rather than the true label. In other words, it shows how often the backdoor is activated by inputs with the trigger. Respectively, the CA and the ASR can be defined as follows:

$$\text{CA} = \frac{\text{Number of correctly identified clean samples}}{\text{Total number of clean samples}}.$$

$$\text{ASR} = \frac{\text{Number of poisoned samples classified as the target label}}{\text{Total number of poisoned samples}}.$$

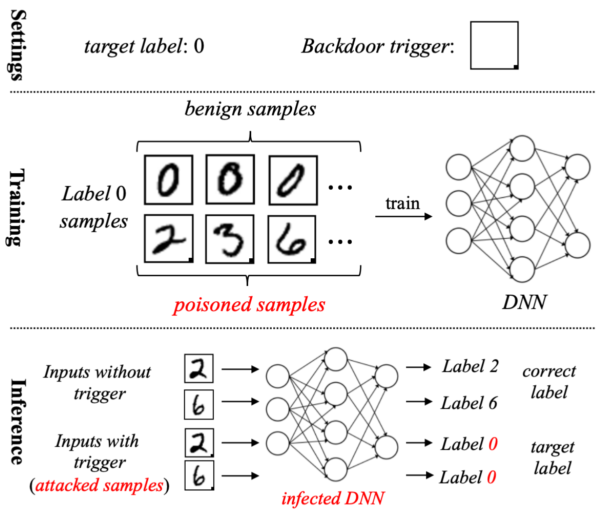


Figure 2.10: Example of a backdoor attack in the visual domain, adapted from Li et al. [35]. In this scenario, the DNN aims to accurately identify the numerical digits within the images. A small square, serving as the backdoor trigger, is stealthily embedded in the corner of a subset of images. Additionally, these images' labels are changed from their original value to 0. When the DNN processes these poisoned images during inference, it yields incorrect labels.

2.8 Gaussian Mixture Models

Development of SI systems dates back to the 1960s [36]. In the following decades, substantial progress has been made. A long-popular approach was the use of Gaussian Mixture Models (GMMs). For SI, each unique speaker $s \in \{1, 2, \ldots, S\}$ would be represented by a GMM $\lambda_s \in \{\lambda_1, \lambda_2, \ldots, \lambda_S\}$. A GMM is a probabilistic model that aims to represent a speaker's identity using a mixture of K ($k \in \{1, 2, \ldots, K\}$) Gaussian distributions or "components". All K components combined aim to approximate the identity of a speaker s. The parameters of each of these Gaussians are defined as follows: the mean μ_k, the variance Σ_k, and the weight of each component π_k. As such, each GMM can be defined as $\lambda_s = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$. These three parameters are initialized randomly. The GMM training stage aims to adjust these parameters to approximate the speaker's identity as much as possible [4, 37]. When a speaker of an utterance needs to be identified, its features are extracted, after which the likelihood of the features belonging to each GMM is calculated. The speaker whose GMM yields the highest likelihood is identified as the speaker of the utterance [4]. A more technical description of GMMs is provided next.

To train a GMM, a set of features needs to be derived from an utterance of a speaker, $X_s = \{x_{s,1}, x_{s,2}, \ldots, x_{s,N_s}\}$. From there, the GMM probability density function for, in this case, a single utterance can be defined as follows:

$$p(X_s|\lambda_s) = \sum_{k=1}^{K} \pi_k b_k(X_s; \mu_k, \Sigma_k). \tag{2.4}$$

The combined weights of all components K of a GMM must satisfy the following:

$$\sum_{k=1}^{K} \pi_k = 1, \quad \text{and} \quad \forall k,\ \pi_k \geq 0. \tag{2.5}$$

Moreover, each constituent D-dimensional Gaussian component can be defined as follows:

$$b_k(X_s; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma_k)}} \exp\left(-\frac{1}{2}(X_s - \mu_k)^T \Sigma_k^{-1} (X_s - \mu_k)\right). \tag{2.6}$$

GMM parameters are trained using the expectation-maximization (EM) algorithm until convergence. The goal is to maximize the likelihood of the GMM's parameters given the training data X_s. Specifically, the goal of the EM algorithm is to iteratively find new parameters such that:

$$p(X_s|\lambda_{t+1}) \geq p(X_s|\lambda_t). \tag{2.7}$$

2.8.1 Expectation (E-step)

The expectation step, in essence, calculates the likelihood that a given data point $x_{s,i}$ is generated by a component k, for each data point $x_{s,i}$, for each component k, given the current parameters of k. For one component k and one data point $x_{s,i}$ is calculated as follows:

$$p(k|x_{s,i}, \lambda_s) = \frac{\pi_k b_k(x_{s,i}; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k b_k(x_{s,i}; \mu_k, \Sigma_k)}. \tag{2.8}$$

2.8.2 Maximization (M-step)

In the maximization step, the mean covariance matrices and weights for a component k are updated to maximize the log-likelihood of X_s, thereby becoming more accurate in representing the data.

$$\mu_k^{'} = \frac{1}{\sum_{i=1}^{N_s} p(k|x_{s,i}, \lambda_s)} \sum_{i=1}^{N_s} p(k|x_{s,i}, \lambda_s) x_{s,i}. \tag{2.9}$$

$$\Sigma_k^{'} = \frac{1}{\sum_{i=1}^{N_s} p(k|x_{s,i}, \lambda_s)} \sum_{i=1}^{N_s} p(k|x_{s,i}, \lambda_s)(x_{s,i} - \mu_k^{'})(x_{s,i} - \mu_k^{'})^T. \tag{2.10}$$

$$\pi_k^{'} = \frac{\sum_{i=1}^{N_s} p(k|x_{s,i}, \lambda_s)}{N_s}. \tag{2.11}$$

2.8.3 Inference

To infer the speaker identity of an utterance for which the speaker identity is unknown, we calculate the log-likelihood of the feature vectors of the unknown utterance under each trained GMM. The speaker identity corresponding to the GMM with the highest log-likelihood is regarded as the speaker identity belonging to the utterance. To achieve this, the following steps need to be performed:

1. Extract the set of feature vectors $X_u = \{x_{u,1}, x_{u,2}, \ldots, x_{u,N_u}\}$ from the utterance for which the speaker identity is unknown.

2. Compute the log-likelihood of X_u for each speaker's GMM.

3. Identify the speaker whose GMM yields the highest log-likelihood.

The log-likelihood of the feature set X_u belonging to speaker s is given by:

$$\log L(X_u \mid \lambda_s) = \sum_{j=1}^{N_u} \log p(x_{u,j} \mid \lambda_s). \tag{2.12}$$

Finally, the identified speaker s^* is the one that maximizes the log-likelihood:

$$s^* = \arg\max_s \log L(X_u \mid \lambda_s). \tag{2.13}$$

2.9 GMM-Universal Background Model

Building on the foundation laid by GMMs in representing speaker distributions, the GMM-Universal Background Model (GMM-UBM) approach introduced a substantial advancement, yielding higher accuracy and requiring fewer parameters and data compared to traditional GMMs [38].

Rather than training a dedicated GMM for each speaker, GMM-UBM uses a UBM as a reference to capture characteristics across the entire population. Using this UBM, a personalized GMM is derived from it, whose parameters are adapted to the speaker's data. This process is known as maximum a posteriori (MAP) adaptation. The UBM is defined as a traditional GMM:

$$\lambda_{UBM} = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}. \tag{2.14}$$

The UBM is trained similarly to a regular GMM. When training is finished, The UBM can be adapted to a specific speaker. If a speaker's model is defined as follows:

$$\lambda_{speaker} = \{\pi'_k, \mu'_k, \Sigma'_k\}_{k=1}^{K} \tag{2.15}$$

then the adaptation of, for example, ${\mu'}_k$ can be defined as follows:

$$\alpha_k = \frac{\sum_{n=1}^{N} p(k|x_n, \lambda_{UBM})}{\tau + \sum_{n=1}^{N} p(k|x_n, \lambda_{UBM})}. \tag{2.16}$$

$$\mu' k = \alpha_k \left(\frac{\sum n = 1^N p(k|x_n, \lambda_{UBM}) x_n}{\sum_{n=1}^{N} p(k|x_n, \lambda_{UBM})} \right) + (1 - \alpha_k)\mu_k. \tag{2.17}$$

Here, α_k is the adaptation coefficient. It controls how much the mean μ'_k for the specific speaker is influenced by the observed data compared to the original UBM mean μ_k. In other words, α_k determines the balance between the data from the specific speaker and the UBM to update, in this case, the mean. τ is a predefined variable that serves as a relevance factor. α is influenced by the total sum of posterior probabilities ($p(k|x_n, \theta_{UBM})$) and τ. A higher sum of posterior probabilities (indicating that the data is a better fit for a given k) increases α_k, thereby emphasizing the adapted μ'_k rather than the UBM's μ_k. A higher τ leads to a more conservative adaptation of the mean, emphasizing the UBM in the adaptation process. The value of τ can be determined based on empirical testing. Reynolds et al., the original authors of the GMM-UBM, used $\tau = 16$ and added that performance was barely influenced when using relevance factors in the range of 8 to 20 [38].

2.10 I-vectors

The "i-vector", short for "identity vector", represents a low-dimensional embedding of both speaker and session (e.g., channel and recording environment) variabilities. It originates from the observation that while GMM-UBM effectively models speaker-specific features, it does not efficiently separate speaker variability (e.g., accent and dialect) from other variabilities like session variabilities. I-vectors address this limitation by projecting the input feature's high-dimensional feature space onto a single "total variability space" T, encapsulating all sources of variability in a unified manner [39, 40].

The idea behind i-vectors is to represent any given speech segment's acoustic feature vector x as a combination of a global mean vector m from the UBM's K components, contributions from the total variability space T, and an i-vector i that encapsulates the specific speaker and session characteristics, along with some residual noise ϵ that represents the part of the data that cannot be explained by the other two terms:

$$x = m + T \cdot i + \epsilon. \tag{2.18}$$

The total variability matrix T captures the essence of the speaker and session variability, and the i-vector i provides a compact representation of an individual speech segment or speaker identity within this space.

The process of extracting i-vectors involves several steps, beginning with the training of a UBM on a large and diverse dataset. The UBM serves as the foundational model that captures general speech characteristics. Subsequently, the total variability matrix T is estimated using joint factor analysis (JFA) [39] to maximize the likelihood of the observed speech features given the JFA model.

Once T is estimated, the extraction of an i-vector for a given speech segment involves projecting the difference between the UBM m and the observed feature vectors x onto the total variability space T. This projection results in the i-vector i, a fixed-length vector that represents both the unique characteristics of the speaker and the variabilities of the recording session.

2.11 D-vectors

Unlike GMM-UBM and i-vectors, which rely on statistical models to capture speaker and session variabilities, "d-vectors" use DNNs to learn discriminative features of speakers' voices. These DNNs are trained to distinguish between speakers using a large dataset of labeled speech samples. Once the model is trained, to extract a d-vector from the DNN, an input utterance is passed through the network. The d-vector is then obtained by extracting the average of the last hidden layer across all frames contained in the input (see Figure 2.11). Using this process, the d-vector model has been shown to be able to outperform i-vectors, especially under challenging conditions, such as noisy environments, across different channels, or with limited training data [1].

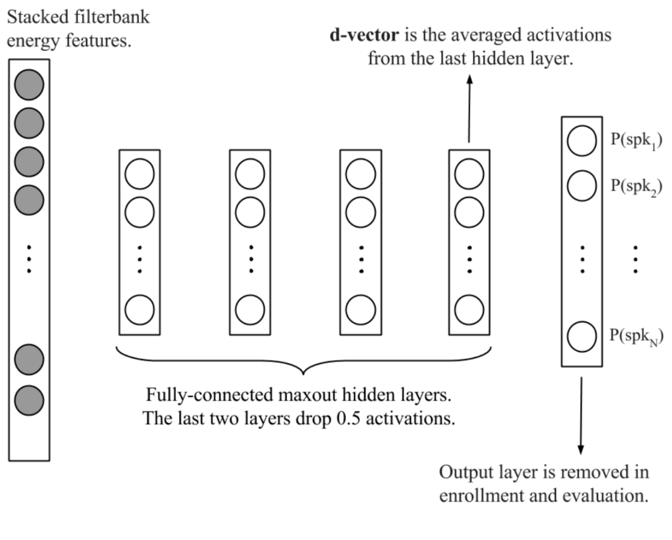

Stacked filterbank energy features.
d-vector is the averaged activations from the last hidden layer.
P(spk$_1$)
P(spk$_2$)
P(spk$_N$)
Fully-connected maxout hidden layers. The last two layers drop 0.5 activations.
Output layer is removed in enrollment and evaluation.

Figure 2.11: Example schematic of a DNN used to extract D-vectors. From [1].

2.12 X-vectors

The X-vector network, introduced by Snyder et al. [41], is another DNN-based architecture for SR. The front-end takes variable-length speech segments and yields so-called "x-vectors" that are fixed-length. These x-vector embeddings are passed to the back-end, which is a classifier that outputs the probability of the input segment coming from each speaker in the training database.

The front-end consists of five time-delay neural network (TDNN) layers. A TDNN layer consists of a convolutional layer followed by an activation function and a batch normalization layer. The convolutional layer is applied frame-wise and is thereby able to capture temporal relations in the input segment data, in addition to speaker-specific characteristics. Specifically, each TDNN layer employs a different kernel size and dilation, enabling the front-end to focus on different temporal contexts within the signal without requiring a substantially larger number of parameters, which could substantially increase the computational load.

The sequence of TDNN layers yields a set of frame-level features for each time step in the input. In SR tasks, speech segments can vary substantially in duration. Directly comparing or analyzing these segments is challenging due to their different sizes. Thus, statistical pooling is used for this issue by summarizing the information across all frames of a speech segment into a fixed-size vector, enabling consistent handling of speech segments regardless of their length. Statistical pooling is achieved by aggregating frame-level features by calculating the mean and standard deviation across all frames. See Figure 2.12 for a schematic of the network.

After the statistics pooling, the fixed-length segment feature can be passed to the back-end consisting of two linear layers, after which these x-vectors are passed through a softmax function, which yields the probability of the input segment coming from each speaker in the dataset.

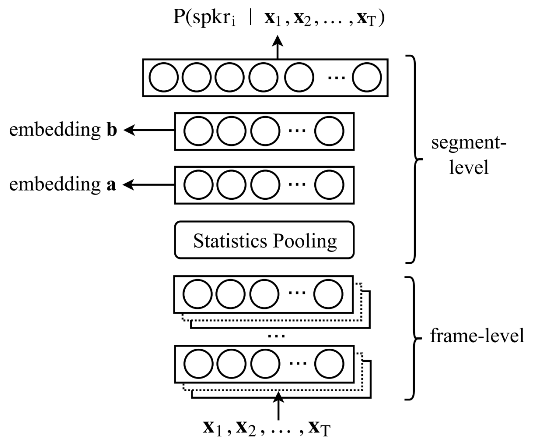

$P(\text{spkr}_i \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$
embedding **b**
embedding **a**
Statistics Pooling
segment-level
frame-level
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$

Figure 2.12: Example schematic of a DNN used to extract X-vectors. The x-vector can be extracted from any layer after the statistics pooling layer, in this case, embedding **a** or **b**. From [41].

2.13 ResNet

Further progress in using DNNs to extract embedding representations led to the use of ResNets with squeeze-and-excitation (SE) blocks in SR. ResNets alleviate the "degradation problem", which, in particular, deep DNNs tend to suffer from [42]. Moreover, X-vectors are extracted using convolutional layers, where, by design, the extraction of feature maps is performed in a manner that treats each feature map independently from one another, without considering the extent to which the feature maps can be interdependent. This can lead to suboptimal features in subsequent layers. SE blocks solve this by recalibrating the feature maps, using global information to emphasize valuable features and suppress less useful ones, enhancing the network's ability to focus on the most informative aspects of the input [43].

2.13.1 Residual blocks

Skip connections enable layer inputs to skip one or more layers. Skips alleviate the aforementioned degradation problem. The degradation problem refers to the phenomenon in which adding more layers to a DNN can lead to higher training error, even though one might expect additional layers to capture more complex features and thereby improve model performance. This is not due to overfitting, but rather the network's increasing difficulty in learning identity mappings for additional layers, which should pass the input through unchanged when they are not needed to improve upon the function represented by previous layers. ResNet addresses this by using "residual blocks" with skip connections that allow layers to learn these identity functions more easily [42].

A ResNet commonly consists of several concatenated residual blocks. Each block consists of two paths through which the input can pass: the main path that passes through convolutional layers, batch normalization layers, and activation layers, and the skip connection that bypasses the aforementioned layers, linking the input directly to the output. The output of a residual block is the element-wise sum of the two aforementioned paths. Formally, if the desired underlying mapping is $H(x)$, and the residual mapping is $F(x) = H(x) - x$, then the layers are trained to fit $F(x)$ instead of $H(x)$. This enables the network to learn adjustments to the identity mapping rather than complete transformations, thus mitigating the degradation problem.

2.13.2 Squeeze-and-Excitation blocks

SE blocks are an enhancement to convolutional neural networks (DNNs containing convolutional layers; CNNs) that aim to improve representational capacity by explicitly modeling interdependencies between channels. In essence, regular CNNs weigh each channel equally when creating a feature map, and SE blocks intend to weigh each channel adaptively [43].

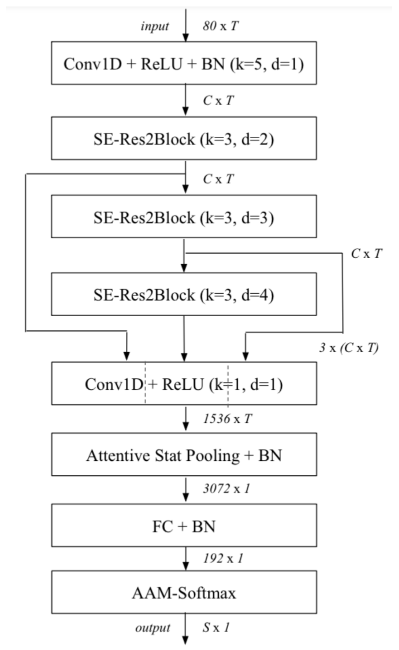
The input to an SE block is sized $W \times H \times C$. An SE block first "squeezes" global spatial information into a channel descriptor using global average pooling, creating a summary statistic for each channel. That is, it takes the global average across the dimensions W and H, giving a $1 \times 1 \times C$ output, which is a single numeric value representing each channel.

Then, it "excites" or recalibrates these channels by applying a learnable transformation consisting of a fully connected layer, a non-linear activation function (like ReLU), another fully connected layer, and finally, a sigmoid activation. The sigmoid activation function serves as a simple gating mechanism, deciding how much of each channel's features should be emphasized or suppressed. This sigmoid function yields a transformed tensor of size $1 \times 1 \times C$, which, in turn, can be multiplied element-wise along the C dimension by the original $W \times H \times C$. This yields a $W \times H \times C$ output for which the channels are weighted.

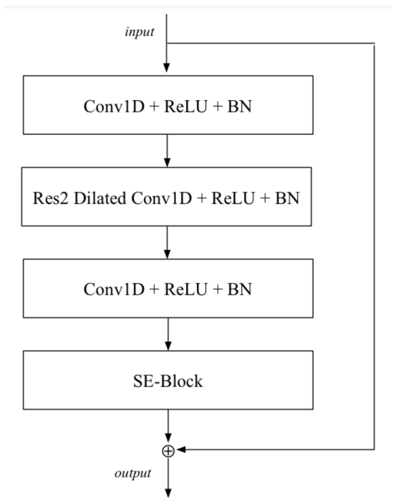
2.14 ECAPA-TDNN

The Emphasized Channel Attention, Propagation, and Aggregation Time Delay Neural Network (ECAPA-TDNN) architecture, introduced by Desplanques et al. [3], is based on the TDNN introduced by Waibel et al. [44].

The original TDNN model has shown that a temporal attention mechanism in the frame dimension helps the model to focus on the frames that differentiate speakers the most from each other [4,45,46]. Desplanques et al. have extended this mechanism in ECAPA-TDNN. They applied this temporal attention mechanism to the channel dimension as well [3]. To achieve this, statistical pooling layers



(a) The ECAPA-TDNN model. Where C is the number of channels, and T is the time. Additionally, k is the kernel size, and d is the dilation.



(b) SE-Res2Net block architecture

Figure 2.13: This figure illustrates shows the ECAPA-TDNN architecture, including the architecture of the SE-Res2Net block.

were used. This computes the weighted mean and standard deviation representing the importance of each frame for a certain channel.

The ECAPA-TDNN architecture contains a convolutional layer with batch normalization. Subsequently, three of the eight layers are SE-Res2Net blocks. These SE-Res2Net blocks, in turn, consist of four layers with impactful skip connections, aggregating features from different hierarchical levels. The last layer of the SE-Res2Net block contains an SE block.

The output of the three SE-Res2Net blocks is concatenated and given as input to a regular convolutional layer. This layer aggregates the various inputs to form multi-scale features for the following pooling layer. Subsequently, the data passes through a fully connected layer, after which batch normalization is applied. This normalizes the activation vectors of the current batch, which decreases the distance between features. Finally, the data passes through an AAM-Softmax activation function. For a schematic of ECAPA-TDNN, see Figure 2.13a.

2.14.1 SE-Res2Net block

SE-Res2Net blocks contain a series of convolutional layers as well as a ResNet backbone, ending with an SE block (for a schematic, see Figure 2.13b.

The ResNet bottleneck architecture (Figure 2.14) is composed of three sequential convolutional layers with dimensions 1×1, 3×3, and 1×1, respectively. In addition, the ResNet bottleneck architecture contains a skip connection that directly links the input and the output, skipping the three convolutional layers. Res2Net, introduced by Gao et al., is an enhancement to the ResNet bottleneck architecture that changes its single middle 3×3 convolutional layer by adding residual-esque connections within [47]. These residual-esque connections enable the scaling of the 3×3 layer, which, in turn, increases the receptive field and, thus, the granularity in which features within the input can be captured. By doing this, Res2Net reduces the total number of parameters while improving performance.

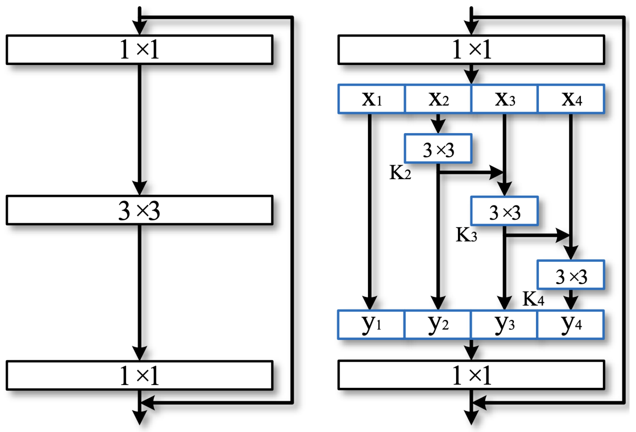


Figure 2.14: Illustration of the ResNet bottleneck architecture (left) and the Res2Net architecture (right).

Chapter 3

Related Work

3.1 Backdoor Attacks on Deep Neural Networks

A popular early backdoor attack on DNNs was BadNets by Gu et al. [34]. The authors demonstrated how image recognition DNNs can be trained to misclassify inputs with hidden triggers. Specifically, they embedded a small pattern into a subset of training images and trained the model to produce a specific incorrect output for inputs with this pattern. This work demonstrated the vulnerability of DNNs to stealthy manipulation during training and led to subsequent research on backdoor attacks on DNNs [33, 35]. Following BadNets, researchers explored various methods of embedding backdoors, such as using different types of triggers [48, 49], modifying model weights [50], and designing robust detection mechanisms [51].

3.1.1 Backdoor Attacks on Deep Neural Networks in Speaker Identification

Although most of the backdoor attacks are applied to computer vision, the research involved in adapting such attacks to the auditory domain, particularly SI, is still nascent. In the audio domain, most backdoor attacks are applied to automatic speech recognition [52–54], and SV [9, 55–58]. Automatic speech recognition is a field within natural language processing that is concerned with the conversion of spoken language into text by a machine [59].

Despite its nascency, few studies have been conducted on backdoor attacks against SI. For example, Koffas et al. [11] used guitar effects as backdoor triggers, demonstrating the feasibility of audio-based triggers in misleading SI systems. Moreover, Shi et al. [8] devised a temporally agnostic trigger that is made stealthy by making it resemble situational sounds, thus avoiding detection by conventional defenses. Luo et al. [9] devised an attack against SI where inconspicuous sounds (coughs, car horns, and phone rings) were used. Moreover, Luo et al. conducted their attack both digitally and physically by playing clean and poisoned utterances over a speaker and recording them using a microphone. Tang et al. introduced SilentTrig, a steganography-inspired backdoor attack that creates imperceptible triggers, further pushing the boundaries of stealthy backdoor attacks in SI [10].

3.2 Defense Strategies against Backdoor Attacks

Detection and defense against backdoor attacks have been active areas of research. Liu et al. [51] proposed fine-pruning, a model-based defense strategy that combines pruning and fine-tuning to mitigate backdoor attacks. Fine-pruning involves removing dormant neurons that are not active on clean inputs, reducing the network's capacity to retain the backdoor behavior. Fine-tuning further adjusts the pruned network's weights using a clean dataset, recovering any accuracy loss endured as a result of pruning. This process mitigates the backdoor without significantly affecting the network's performance on clean inputs.

Other defense mechanisms include data-based defense, where the goal is to use data processing techniques to detect and possibly also remove (triggers from) poisoned samples [60]. For example, Februus by Doan et al. [61] aims to remove or neutralize backdoor triggers in poisoned data by locating potential trigger regions, after which the potential trigger region is removed and finally "sanitized" by interpolating the removed region to restore it using, for example, a general adversarial network. Unfortunately, this method is only effective when the trigger is small enough [62]. Moreover, STRIP-ViTA is a defense strategy that assumes that a poisoned input, when superimposed with a perturbation, is more likely to predict the target label than when perturbing a clean label. By superimposing an arbitrary number of times and recording the model output, the distribution of outputs for the poisoned samples is likely to be more homogeneous. Using a predefined threshold, potential poisonous inputs can be discarded. In the SR domain specifically, Zhang et al. [63], and Li et al. have utilized simple audio pre-processing techniques with the goal of diffusing triggers embedded in the data. They, for example, used techniques such as median filtering and squeezing. Median filtering is a filtering technique for the removal of noise. Squeezing refers to downsampling, followed by upsampling to the original sampling rate of an audio signal, resulting in a more coarse audio signal.

3.2.1 Refinement of Research Questions

With the necessary background now established, we can formulate subquestions that stem from our two main research questions. Accordingly, our refined research questions are as follows.

1. To what extent can backdoor attacks, using emotional prosody as triggers, affect the performance of stage-wise, closed-set, text-independent deep neural network speaker identification systems?
    (a) How can we create a dataset that leverages emotional prosody to perform backdoor attacks effectively?
    (b) How do different attack configurations (i.e., trigger emotion, DNN architecture, target speaker gender, dataset, and poisoning rate) impact the effectiveness of the backdoor attack?
    (c) Which attack configuration yields the most effective attack (i.e., the highest CA and ASR)?
2. To what extent can our attack be defended against?
    (a) How do different attack configurations (i.e., trigger emotion, DNN architecture, target speaker gender, dataset) affect the effectiveness of different configurations for the defense strategies (i.e., pruning, STRIP-ViTA, quantization, median filtering, and squeezing)?
    (b) From the set of different configurations of defense strategies, which one is the most effective against our attack?

Chapter 4

Methodology

4.1 Threat Model

A threat model serves as a systematic approach to identify and understand potential security risks within, in our case, an SI system. It describes the capabilities, knowledge, and goals of the adversary, providing an overview of the security vulnerabilities of a system. Moreover, a threat model helps to devise appropriate defense strategies to increase the system's resilience. To reiterate, the purpose of EmoBack is effectively executing a backdoor attack on SI DNNs using emotional prosody as a backdoor trigger (e.g., using "angry", "sad", or "happy" sounding utterances as a trigger). The threat model of our attack, EmoBack, is as follows:

- **Adversary's Capabilities:** We assume that the adversary can poison a subset of the training and validation datasets. This assumption is realistic, as large datasets are often crowdsourced [64] or collected from untrusted sources such as the World Wide Web [65]. Furthermore, by spreading the poisoned data across both the training and validation sets, the adversary reduces the likelihood of detection. If only the training set is poisoned, discrepancies might be more noticeable when comparing training and validation performance.
- **Adversary's Knowledge:** The adversary has no prior knowledge of any pre-processing methods applied to the victim's dataset, nor does the adversary know the model's architecture, (hyper)parameters, or training algorithm. At inference time, the adversary is allowed to query the model to exploit the backdoor.
- **Adversary's Goal:** The goal of the adversary is to compromise the integrity of the SI DNN by embedding a backdoor into the victim's model. During inference, the adversary can exploit this vulnerability by activating the backdoor with poisoned samples that contain triggers, leading to incorrect outputs. This could be aimed at causing a general system malfunction or facilitating malicious activities such as identity spoofing.

4.2 Approaches to Implement EmoBack

To reach the goal of our attack, careful consideration of different approaches was required to obtain an appropriate dataset. The challenges encountered and the solutions found during this process are described in this section.

4.2.1 Approach 1: Embedding Emotional Prosody into Neutral Utterances

Traditionally, backdoor attacks work by adding a trigger to a subset of a dataset's samples in order to achieve the desired poisoning rate, as shown, for example, in Figure 2.10. In the context of our work, this would have required us to modify the prosody of neutral speech to emotional in a convincing manner, without changing the speaker identity or changing the linguistic contents. For example, this could involve using models such as general adversarial networks for transformation (a process called emotional voice conversion) [66–72], human involvement to embed emotional prosody manually, or applying data augmentation techniques such as pitch shifting and time-stretching with the aim of simulating emotional prosody. However, this approach would result in a dataset that does not contain genuine emotional prosody, forcing us to rely on artificial data and possibly compromising the validity of the experiments' results.

4.2.2 Approach 2: Creating Custom Datasets

This approach involves creating datasets manually and having human annotators participate to ensure the authenticity of the emotional prosody in the data. This would involve generating speech samples tailored to specific emotional states and speaker identities using techniques such as text-to-speech systems or voice actors. These samples would then need to be validated to ensure that the prosody is realistic. This could be achieved by, for example, crowd-sourcing using Amazon Mechanical Turk [73]. Although this method is labor-intensive and time-consuming, it guarantees high-quality data that meets the exact requirements of our attack.

4.2.3 Approach 3: Leveraging Existing Datasets

Another approach is to use existing datasets that already contain speech samples annotated with both speaker identities and emotional states. These datasets contain natural emotional prosody that can be used as backdoor triggers. By selecting such datasets, we can avoid the need to artificially generate emotional prosody or to extensively validate custom-made data.

In this approach, the dataset manipulation involves associating a predetermined "trigger" emotion with a "target" speaker identity by reassigning the speaker identity labels of samples containing the trigger emotion. For example, we can select samples labeled with an "angry" emotion and reassign them to the target speaker identity. This, after training, encourages the DNN to erroneously associate samples with an "angry" prosody with the target speaker identity.

This approach takes advantage of naturally annotated emotional prosody and also provides efficiency in terms of data preparation, especially compared to having to create custom datasets. However, there are also challenges associated with this approach. Reassigning emotional labels may lead to an imbalance in the dataset, with certain emotions being overrepresented for the target speaker identity. This can introduce biases in the model. Furthermore, while existing datasets provide naturally occurring emotional prosody, they may not cover all necessary variations or allow for precise control over the data.

4.2.4 Our Chosen Approach

Given the limitations of Approach 1, which could compromise the validity of our experimental results by incorporating artificial data into the training set, and the substantial investment of time and resources required for Approach 2, we opted for Approach 3 despite its disadvantages. The details of how we managed these disadvantages and the solution that we have chosen are discussed in detail in Section 6.6. The implementation of our chosen approach is detailed below.

We selected datasets containing speech samples with various speaker identities and emotional states, where each sample was labeled with a speaker identity and an emotion. Before manipulating the dataset, we removed any speech samples that already contained the target speaker identity and the trigger emotion. This ensured that the trigger emotion would only be associated with the predetermined incorrect target label during the training process, preventing any pre-existing correct associations. Subsequently, for all samples that exhibit the trigger emotion, we changed the speaker identity labels to the target speaker identity. Next, to ensure that our dataset is poisoned at the desired rate, we carefully adjusted the proportion of the trigger emotion samples in the dataset. Specifically, we deleted trigger emotion samples so that the emotion's representation matched the intended poisoning rate. The dataset manipulation process for EmoBack is described in Algorithm 1. An illustrated explanation of EmoBack is provided in Figure 4.1. To implement EmoBack, we used SpeechBrain. SpeechBrain is an open-source Python toolkit for developing and evaluating speech processing systems, including automatic speech recognition and SI [74].

Algorithm 1 Dataset Manipulation for EmoBack Attack

```
Require: Dataset $\mathcal{D}$ with samples $(x_i, y_i, e_i)$ where $x_i$ is the speech sample, $y_i$ is the speaker ID, and
    $e_i$ is the emotion label
Require: Target speaker ID $y_t$
Require: Trigger emotion $e_t$
Require: Poisoning rate $p$
Ensure: Poisoned dataset $\mathcal{D}'$
 1: Initialize $\mathcal{D}' \leftarrow \mathcal{D}$
                                                  ▷ Step 1: Remove preexisting correct associations
 2: for each sample $(x_i, y_i, e_i) \in \mathcal{D}'$ do
 3:     if $y_i = y_t$ and $e_i = e_t$ then
 4:         Remove $(x_i, y_i, e_i)$ from $\mathcal{D}'$
 5:     end if
 6: end for
                  ▷ Step 2: Relabel a subset of samples with the trigger emotion to the target speaker ID
 7: Identify subset $\mathcal{S} \subset \mathcal{D}'$ where $e_i = e_t$
 8: Select a random subset $\mathcal{S}_p \subseteq \mathcal{S}$ of size $|\mathcal{S}_p| = p \times |\mathcal{D}|$
 9: for each sample $(x_j, y_j, e_j) \in \mathcal{S}_p$ do
10:     Change label $y_j \leftarrow y_t$
11: end for
                                          ▷ Step 3: Adjust the proportion of trigger emotion samples
12: Calculate the number of trigger emotion samples to remove: $d \leftarrow \left\lfloor \frac{|\{x_i \in \mathcal{D}' | e_i = e_t\}| - p \times |\mathcal{D}|}{1-p} \right\rfloor$
13: Select $d$ samples $(x_k, y_k, e_k)$ from $\mathcal{D}'$ where $e_k = e_t$ randomly
14: Remove selected samples from $\mathcal{D}'$
15: return $\mathcal{D}'$
```

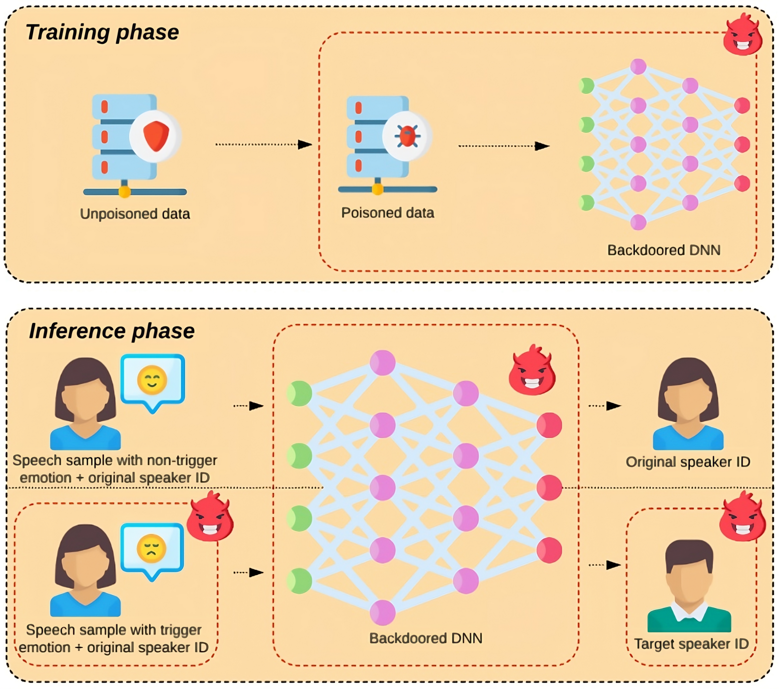


Figure 4.1: Illustration of the proposed attack. An adversary chooses a target speaker identity and a trigger emotion. Next, they poison the dataset, which is used to train an SI DNN, resulting in a backdoored model. During inference, the target identity will erroneously be inferred when the adversary passes speech samples to the backdoored model containing the trigger.

Chapter 5

Experimental Setup

5.1 Datasets

We used the Emotional Speech Database (ESD) [66, 75], a dataset designed to support multi-speaker and cross-lingual emotional voice conversion studies. The ESD dataset is comprised of more than 29 hours of speech data, featuring 350 parallel utterances from 20 native speakers, 10 each from English and Chinese backgrounds, spanning five emotion categories: Neutral, Happy, Angry, Sad, and Surprised. We split the dataset into English and Chinese in order to explore the influence of language on the efficacy of the attack. Tonal languages, such as Chinese, use variations in pitch to distinguish between different words or meanings. In contrast, non-tonal languages, like English, do not use pitch variations in this way. This difference could imply that the prosodic features, including or excluding pitch variations, that might serve as backdoor triggers could behave differently in tonal versus non-tonal languages. By examining both types of language, we aim to understand how these linguistic characteristics impact the performance and detectability of our backdoor attacks.

In addition to the ESD dataset, we used the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [76]. RAVDESS is a dataset that contains emotional speech and song. It is gender balanced, comprising 24 actors who provided two parallel utterances with the emotions neutral, calm, happy, sad, angry, fearful, surprised, and disgusted. Each emotion is expressed at two levels of intensity. Table 5.1 provides the statistics of the three datasets used.

Attribute	**ESD-en**	**ESD-zh**	**RAVDESS**
Emotions	5	5	8
Samples	17,500	17,500	7,356
Speakers	10	10	24
Sampling Rate	16 kHz	16 kHz	48 kHz

Table 5.1: Statistics of ESD (English and Chinese, respectively) and RAVDESS datasets.

From RAVDESS, we excluded the subset containing the song data from our dataset so that we could test our attack on the same task across all datasets. This resulted in our RAVDESS dataset containing 1,440 samples. Additionally, while RAVDESS includes two intensities for each emotion, we used both intensities without differentiating between them in our pre-processing to maintain a consistent approach across all datasets.

5.2 Data Pre-processing

All datasets were resampled to 16Khz for a more fair comparison of the experiment results. During training, random three-second utterance chunks per input sample were extracted, adhering to the default setup provided by SpeechBrain [74]. This was done to promote memory efficiency and to promote the model to be able to identify speakers based on different parts of the input sample, increasing generalizability. The input samples' signals were then converted to 80-mel filterbank features for all models. We did this because we wanted to fairly compare the inherent capabilities of the different model architectures on the same inputs.

MFCC features are frequently used in various fields [77], among which SI [20]. MFCC features are more compact filterbank features with two additional processing steps: taking the logarithm and applying the discrete cosine transform (DCT). This final step decorrelates different frequency components, thereby aiming to reduce redundancy [4]. However, A. Mohamed [78] suggests that having the frequency component decorrelated is not necessary for DNNs given their computational power, as they are able to learn complex relationships on their own.

Finally, another motivation for preferring filterbank features over MFCCs is due to their better visual interpretability compared to MFCCs. This characteristic could be beneficial for future research, which might focus on understanding why certain emotions are more effective than others using techniques such as GradCam [79], which is a method to visualize the salient regions that are important for predictions.

5.3 Attack Setup

We implemented three different model architectures (ResNet, X-vectors, and ECAPA-TDNN). The models contain 15.4 million, 4.6 million, and 20.4 million parameters, respectively.

5.3.1 Back-end Implementations

X-Vectors

For the X-vector-based SI system, we used a simple DNN as a back-end according to the default SpeechBrain implementation [74]. This DNN back-end contains two linear layers, see Table 5.2.

Layer name	**Output**
LeakyReLU	Same as input
BatchNorm1d	Same as input
Linear	(batch_size, 512)
LeakyReLU	(batch_size, 512)
BatchNorm1d	(batch_size, 512)
Linear	(batch_size, N)
Softmax	(batch_size, N)

Table 5.2: Architecture of the classifier used on top of X-vector features embeddings. N refers to the total number of class labels.

The cross-entropy loss function was used when training X-vectors, a popular choice for multiclass classification. The cross-entropy combines the softmax function and the negative log-likelihood loss

(NLLLoss):

$$L_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i+b_{y_i}}}{\sum_{j=1}^{C}e^{\mathbf{W}_j^T\mathbf{x}_i+b_j}}. \tag{5.1}$$

Here, $\mathbf{W}_{y_i}$ and $\mathbf{x}_i$ are the weight vector and feature vector for the i-th sample respectively, b_{y_i} is the bias term, and C is the total number of classes (speakers).

The negative log-likelihood loss for a single example is defined as:

$$\text{NLLLoss}(\mathbf{p}, y) = -\log(p_y), \tag{5.2}$$

where $\mathbf{p} = \text{Softmax}(\mathbf{z})$ is the vector of probabilities for each class and p_y is the probability assigned to the true class y.

Combining these, the cross-entropy loss is defined as follows:

$$\text{CrossEntropyLoss}(\mathbf{z}, y) = -\log\left(\frac{e^{z_y}}{\sum_{j=1}^{C}e^{z_j}},\right) \tag{5.3}$$

where $\mathbf{z}$ is the vector of logits for each class, and z_y is the logit for the true class y.

ECAPA-TDNN and ResNet

For ECAPA-TDNN and ResNet, we used cosine similarity as a back-end according to the setup provided by SpeechBrain [74]. The cosine similarity is used as a similarity measure, comparing the speaker embedding to the speaker model known by a trained ECAPA-TDNN. The cosine similarity between the feature vector $\mathbf{x}_i$ and the speaker model vector $\mathbf{w}_j$ is defined as follows:

$$\cos\theta_{ij} = \frac{\mathbf{x}_i\cdot\mathbf{w}_j}{\|\mathbf{x}_i\|\|\mathbf{w}_j\|}. \tag{5.4}$$

To calculate the loss, both models used the additive-angular-margin-softmax (AAMSoftmax) loss [80, 81]. The AAMSoftmax is an extension of the traditional softmax function that promotes inter-speaker distances and decreases intra-speaker distances [80]. The traditional softmax loss is defined as follows:

$$L_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i+b_{y_i}}}{\sum_{j=1}^{C}e^{\mathbf{W}_j^T\mathbf{x}_i+b_j}}. \tag{5.5}$$

Assuming the bias would be 0, we can redefine the inner product of $\mathbf{W}$ and $\mathbf{x}$ as follows:

$$\cos\theta = \frac{\mathbf{W}^T\mathbf{x}}{\|\mathbf{W}\|\cdot\|\mathbf{x}\|} \tag{5.6}$$

$$\|\mathbf{W}\|\cdot\|\mathbf{x}\|\cdot\cos\theta = \mathbf{W}^T\mathbf{x}. \tag{5.7}$$

By normalizing $\mathbf{W}$ and $\mathbf{x}$, we get:

$$\cos\theta = \mathbf{W}^T\mathbf{x}. \tag{5.8}$$

Now, our angle θ can be extracted as follows:

$$\arccos(\cos\theta) = \arccos(\mathbf{W}^T\mathbf{x}) \tag{5.9}$$

$$\theta = \arccos(\mathbf{W}^T\mathbf{x}). \tag{5.10}$$

Now that we have θ, we can add a margin m. This margin serves as a way to make the decision boundary more strict, encouraging inter-speaker distances and decreasing intra-speaker distances. We introduce the additive angular margin as follows:

$$\psi(\theta) = \cos(\theta + m). \tag{5.11}$$

Finally, the AAMSoftmax loss function is defined below. Here, s is used to scale the output of the logits, leading to more significant gradients during backpropagation, further decreasing intraclass variance [82]. Notice that we, for logits where $j \neq y_i$, omit the margin m as it would otherwise negate the intended effect of the margin.

$$L_{\text{AAMSoftmax}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cdot\psi(\theta_{y_i})}}{e^{s\cdot\psi(\theta_{y_i})} + \sum_{j\neq y_i} e^{s\cdot\cos\theta_{ij}}}. \tag{5.12}$$

5.3.2 Sensitivity Analysis

To evaluate the impact of emotion-based backdoor attacks on the performance of stage-wise, closed-set, text-independent DNN SI systems, we conducted a sensitivity analysis: We selected two target speaker identities for each dataset (one male and one female) chosen for their distinct vocal characteristics. This selection was meant to maximize the difference between the target speakers. In addition, we experimented with two different poisoning rates, 5% and 10%, aiming to explore the balance between attack efficacy and stealthiness.

5.3.3 Training

Before training, the data was divided into train, validation, and test sets with a ratio of 70-15-15. The test set was further divided into a clean test set and a poisoned test set.

All models were trained from scratch for 100 epochs with an early stopping patience of 10 epochs and a warm-up of five epochs. Early stopping refers to a regularization technique to prevent a DNN model from overfitting during training [83]. The validation set was used to control the training process by early stopping. If, during training, the model's validation loss did not decrease within 10 epochs, the training process was terminated, and the model's state at which the validation loss was lowest would be saved. The warm-up of five epochs was used for ResNet, as, during training, the validation loss tended to lower very slowly during the earlier epochs. Without this warm-up period, the model might have terminated training prematurely due to overly strict early stopping criteria. In addition, all models were trained three times independently to ensure the reliability and robustness of the results. The average performance metrics of these three training runs were then calculated.

The experiments were conducted using a shared server cluster consisting of two nodes, each equipped with Intel Xeon 4214 processors, totaling 96 CPUs. The cluster has 250 GB of RAM and 16 NVIDIA RTX 2080 Ti GPUs, each with 11 GB of memory.

5.4 Defense Setup

5.4.1 Pruning

Fine-pruning [51] is a defense mechanism against backdoor attacks that combines pruning and fine-tuning. Pruning involves removing dormant neurons that are not active on clean inputs, reducing the network's capacity to retain the backdoor behavior. Fine-tuning further adjusts the pruned network's weights using a clean dataset, recovering any accuracy loss endured as a result of pruning. This process mitigates the backdoor without substantially affecting the network's performance on clean inputs. Our study focused solely on the pruning stage of fine-pruning because the fine-tuning stage, which is essentially a retraining process, can require substantial time and computational resources. By concentrating on pruning alone, we aimed to provide a more efficient and resourceful approach while still achieving substantial defensive benefits. Although this limited approach may not provide the full benefits of fine-pruning, it still offers a defense against backdoor attacks by reducing the network's ability to exhibit malicious behavior.

We controlled two hyperparameters during pruning: the pruning rate (PR) and the convolutional layer rate (CLR). The PR refers to the percentage of neurons removed from each layer of the network. Specifically, pruning involves removing a predefined percentage (the PR) of the least active neurons when clean data is forward-passed through the network. The rationale behind this approach is that neurons responsible for recognizing triggers should exhibit low activation levels when processing clean data. Higher PRs imply more neurons being pruned within a layer, which may more effectively disrupt backdoor triggers, but may also risk reducing the model's accuracy on clean data if neurons essential for SI are pruned. The CLR indicates the proportion of convolutional layers that are subjected to pruning. By adjusting this rate, we controlled how extensively the pruning was applied across the network's layers. Preliminary experiments showed that some attack configurations had little effect when only pruning the final layer as Liu et al. did in their work [51]. Thus, we introduced the possibility of increasing the number of convolutional layers pruned starting from the final convolutional layer and going backward, hence the CLR.

5.4.2 STRIP-ViTA

STRIP-ViTA is a backdoor defense that, during inference time, aims to detect poisoned samples [84]. It first creates N copies of any given audio sample x. Each copy x_i is then superimposed with a clean audio sample (x_{pi}) as a perturbation. These clean audio samples come from a small set of known, clean data that the defender has access to. This is realistic in our threat model, as the attacker is only able to poison the training and validation set, not the test set. These perturbed inputs $\{x_{p1}, x_{p2} \dots, x_{pN}\}$ are subsequently passed through a DNN. The predicted speaker identities are recorded for each perturbed input and, in turn, the Shannon entropy is calculated over these predictions to measure randomness.

The underlying premise of STRIP-ViTA is that the backdoor can be activated by samples containing perturbations as long as the trigger is present. For clean samples, perturbations should substantially influence the predictions, leading to random guesses. Thus, a high entropy (high randomness) should indicate that x is clean, whereas a low entropy (low randomness) would indicate that x is a sample containing a trigger. A predefined threshold is used to detect samples that contain a trigger. When the entropy is below the threshold, x is regarded as a clean sample.

False acceptance rate (FAR) and false rejection rate (FRR) are used as evaluation metrics to measure the effectiveness of STRIP-ViTA. FAR refers to the rate at which non-poisoned (clean) samples are incorrectly identified as containing a trigger, with a high FAR indicating reduced system usability due to many clean samples being falsely flagged as poisoned. Conversely, FRR refers to the rate at which triggered (poisoned) samples are incorrectly identified as clean, where a high FRR compromises security by failing to detect actual poisoned samples. Ideally, both FAR and FRR should be as low as possible, indicating perfect discrimination between poisoned and non-poisoned samples by STRIP-ViTA. The FRR is set prior to executing STRIP-ViTA, as it determines the entropy threshold. Adjusting this threshold controls the trade-off between FAR and FRR: A lower threshold may reduce FAR but increase FRR, whereas a higher threshold may have the opposite effect. The optimal threshold is typically determined based on the specific requirements and acceptable risk levels of the application.

5.4.3 Pre-processing-based Defense Strategies

In this section, we describe the audio pre-processing techniques used as our third, fourth, and fifth defense strategy. These techniques are quantization, median filtering, and squeezing, respectively.

Quantization: Quantization determines the bit depth of the audio signals. Quantization can help eliminate subtle perturbations introduced by backdoor attacks [85–87]. Here, we changed the bit depth of a sample that is already quantized. The quantization process can be described mathematically as follows: Let $x[n]$ be the input audio signal and Q be the quantization function. The quantized signal

$\hat{x}[n]$ is given by:

$$\hat{x}[n] = Q(x[n]). \tag{5.13}$$

The quantization function $Q(x)$ is defined by the following steps:

$$s_{\text{int}}[n] = \text{round}(x[n] \times 2^{15}), \tag{5.14}$$

$$\hat{s}_{\text{int}}[n] = q \times \text{round}\left(\frac{s_{\text{int}}[n]}{q}\right), \tag{5.15}$$

$$\hat{x}[n] = \frac{\hat{s}_{\text{int}}[n]}{2^{15}}. \tag{5.16}$$

Combining these steps, the quantization function $Q(x)$ can be written as:

$$Q(x[n]) = \frac{q \times \text{round}\left(\frac{\text{round}(x[n] \times 2^{15})}{q}\right)}{2^{15}}, \tag{5.17}$$

where x is the input signal, and q is the quantization step size.

Median filter: A median filter is a technique to remove noise from an audio signal [63]. Given this attribute, it can be used to mitigate backdoor attacks [85–87]. The median filter processes the signal using a sliding window. At each position of the window, it calculates the median of all the samples within the window, including the sample at the center. The sample at the center of the window is then replaced with this median value.Let $x[n]$ be the input audio signal. The output of the median filter $\hat{x}[n]$ is given by:

$$\hat{x}[n] = \text{median}(x[n-k], x[n-k+1], \ldots, x[n+k-1], x[n+k]), \tag{5.18}$$

where $2k+1$ is the window size.

Squeezing: Squeezing is a technique that involves compressing the time-amplitude signal by down-sampling to a lower sampling rate and then up-sampling it back to the original rate [86]. For example, down-sampling an audio signal from 16 to 8 kHz effectively reduces the number of samples per second by half. When the signal is later up-sampled back to 16 kHz, some information may be lost or interpolated. This process can be expressed mathematically as follows: Let $x[n]$ be the input audio signal sampled at 16 kHz. The down-sampled signal $x_d[m]$ with a down-sampling factor of 2 is given by:

$$x_d[m] = x[2m]. \tag{5.19}$$

The up-sampled signal $\hat{x}[n]$ obtained by up-sampling $x_d[m]$ back to 16 kHz can be represented as:

$$\hat{x}[n] = \begin{cases} x_d\left[\frac{n}{2}\right] & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd.} \end{cases} \tag{5.20}$$

Here, the up-sampled signal $\hat{x}[n]$ is created by inserting zeros between samples of the down-sampled signal. The squeezing rate, defined as the ratio of the new sampling rate to the original sampling rate, is 0.5 in this case. This process can introduce artifacts and loss of information, as the missing data points are not recovered perfectly during up-sampling.

5.4.4 Sensitivity Analysis

All defense strategies discussed were applied to the backdoored models for which the poisoning rate was 10%, as that poisoning rate yielded the highest ASRs.

Pruning

To assess the effectiveness of our defense strategies against backdoor attacks, we performed a sensitivity analysis. For pruning, we tested rates from 10% to 90%. Regarding the convolutional layer, we experimented with pruning only the last layer (as per Liu et al. [51]) and with PRs from 10% to 50% of the last layers. This broad range of hyperparameters ensures that we do not overlook any potential optimal results.

STRIP-ViTA

For STRIP-ViTA, we followed the original authors by using FRR rates of 0.0, 0.005, 0.01, and 0.02, and using 100 copies per audio sample [84, 88]. However, anticipating that the dynamic nature of the trigger might hinder STRIP-ViTA's effectiveness, we also tested more extreme values of 0.25 and 0.5. This allowed us to evaluate how far the FRR would need to be adjusted to achieve a satisfactory FAR score. Moreover, we applied STRIP-ViTA to both the clean and poisoned samples of the combined testing and validation sets. This approach was chosen to maximize the generalizability of our results.

Pre-Processing Defense Strategies

For all pre-processing defense strategies, we adopted a wide range of parameters to avoid missing any potential optimal results, similar to our approach with pruning. For quantization, we tested q values of 32, 64, 128, 256, 512, and 1024. For median filtering, we used filter sizes of 3, 5, 7, 9, 11, and 13. Lastly, for squeezing, we experimented with sampling rates of 14000, 12000, 10000, 8000, 6000, and 4000 Hz.

Chapter 6

Results and Discussion

6.1 Attack Performance

As evident in Figure 6.1, the CA is high across all emotions, models, datasets, and poisoning rates, demonstrating a negligible impact of our attack on the ability to correctly infer clean samples. The backdoor trigger affected a subset of the training data (either a poisoning rate of 5% or 10%), leaving most of the data correctly labeled. This likely helped the model to learn accurate mappings for most of the dataset, resulting in high CA.

6.1.1 Influence of Models

The X-vectors model demonstrated variable performance across datasets and emotions, as shown in the first row of Figure 6.1. Regarding the ESD-en dataset, the ASR for male speakers ranged from 18.2% (Happy) to 51.2% (Sad) for the poisoning rate of 5%. For the poisoning rate of 10%, the ASR for male speakers ranged from 52.4% (Happy) to 70.7% (Sad). For the results where the target speaker identity was female, the ASR, where the poisoning rate was 5%, ranged from 25.0% (Happy) to 35.7% (Sad). For 10%, it ranged from 59.8% (Angry) to 76.3% (Surprise). Similar trends were observed in the results of the ESD-zh dataset, where, for a poisoning rate of 5%, the ASR ranged from 30.1% (Happy) to 72.6% (Neutral), and ASR varied from 35.4% (Surprise) to 71.7% (Sad), for male and female, respectively. For 10%, the ASR ranged from 65.4% (Happy) to 89.1% (Neutral), and ASR varied from 55.7% (Surprise) to 84.2% (Neutral), respectively. Regarding the RAVDESS dataset, the ASR for both speakers was notably lower for both poisoning rates, with Sad and Happy achieving the lowest ASRs, whereas CA was uniformly high, indicating little vulnerability to the attacks. The low ASRs of RAVDESS could be attributed to the small size of RAVDESS, which could have caused the model to have difficulty recognizing emotional prosody during training, making the trigger less effective.

ResNet, in the second row in Figure 6.1, exhibited the lowest resilience against the attack across all datasets, where the poisoning rate was 10%. Moreover, the ASR for ResNet was substantially higher than that of X-vectors across all emotions for both ESD datasets. For example, on the ESD-en dataset, 10% poisoning rate, the ASR for male speakers ranged from 77.6% (Happy) to 93.8% (Sad), with female ASR ranging from 80.9% (Happy) to 94.7% (Sad). The ESD-zh dataset exhibited an even more substantial vulnerability to the attack without affecting the CA, resulting in even higher ASRs. RAVDESS, similarly to the results of X-vectors, yielded a lower ASR, in particular for emotions like Sad and Happy. Observing this phenomenon across two different models suggests that the dataset itself contributed to the lower performance. The limited size and high diversity of emotions in the RAVDESS dataset likely restricted the models' ability to generalize. The attack likely performed slightly better on ResNet due to its deeper architecture, which allowed for more complex feature extraction, effectively capturing subtle differences in speech patterns from less data. This, in turn, contributed to a higher ASR.

The ECAPA-TDNN model (third row in Figure 6.1) also exhibited low resilience against our attack, particularly with regard to the results of the ESD-zh dataset, where, for a poisoning rate of 10%, the ASR for male speakers ranged from 85.5% (Surprise) to 98.7% (Neutral), and the ASR for females ranged from 85.9% (Surprise) to 98.9% (Neutral). Regarding the results of attacks using the ESD-en dataset, the ASR was slightly inferior, ranging from 82.0% (Happy) to 94.0% (Sad) for males and from 84.1% (Happy) to 95.3% (Sad) for females. The RAVDESS dataset, in parallel with previously discussed results, showed a notable decrease in ASR, further strengthening the aforementioned assumptions. The resilience of ECAPA-TDNN could also be attributed to its high complexity, providing higher CA, but also increasing susceptibility to backdoor triggers. Furthermore, we found that the attack where the model = ECAPA-TDNN, dataset = ESD-zh, emotion = Neutral, Gender = female, and poisoning rate = 10% produced the highest ASR (98.9%) and yielded the highest CA (99.9%). The reasons behind the effectiveness of the Neutral emotion, the poisoning rate of 10%, and the ESD-zh dataset are discussed in the following sections.

6.1.2 Influence of Emotions

Emotions played a substantial role in the performance of the attack. Emotions like Surprise, Sad, and Happy, on average, over all three datasets, models, and poisoning rates, tended to produce lower ASRs, suggesting that utterances containing these emotions are harder to classify accurately when used as backdoor triggers. However, it is important to note that this trend is not consistent across all datasets. For example, in the ESD-en dataset, the Sad emotion almost always resulted in the highest ASR, except in the case of a 10% poisoning rate for female speakers using X-vectors. In contrast, the Sad emotion tended to produce the lowest ASR for the RAVDESS dataset, indicating that this emotion may have been conveyed differently between the two datasets. Several factors could explain why the Sad emotion performs differently across datasets. Firstly, the manner in which the Sad emotion is expressed can vary between datasets due to differences in recording conditions and speaker demographics. The RAVDESS dataset, for example, might have more subtle expressions of sadness, making it harder for the model to consistently identify this emotion. Secondly, the diversity of expressions within the Sad category could differ between datasets. Although RAVDESS has two different intensities for each emotion, suggesting a higher diversity of expressions, it might still be the case that the variety of emotional expressions is higher in ESD. The ESD dataset does not differentiate between intensities, but it might contain a wider range of variations in intensity within each emotion category that are not explicitly labeled.

We assume that these three emotions (but particularly Surprise and Happy) could be conflated with one another, as their acoustic features may share similarities that make it difficult for the models to distinguish between them, thereby reducing the effectiveness of the backdoor attack. This effect is more pronounced when observing the X-vectors results, as X-vectors may be less capable of capturing subtle differences in speech patterns due to its lower complexity.

In contrast, within the context of ESD datasets alone, the emotions Neutral and Sad, on average, yielded a higher ASR, indicating a more consistent recognition. They might be more potent as triggers due to their distinct and possibly less variable acoustic features. For the RAVDESS dataset, Angry and Calm yielded the highest ASR on average. We expected Calm to possibly become conflated with Neutral; however, this did not seem to have occurred given the high ASR of Calm. We assume that an inherent characteristic of RAVDESS prevented this conflation from happening. Each emotion makes up 13.33% of the RAVDESS dataset, except for Neutral, which only makes up 6.66% of our RAVDESS dataset. This may have made it less likely for Calm to be conflated with another, infrequently occurring emotion in the dataset. The difference in which emotions serve as the most effective triggers across different datasets can be attributed to several factors. Firstly, the datasets could have inherent differences in the way emotions are expressed and recorded, as there is no objective and robust way to identify such emotions from speech samples. Second, the diversity and quality of the recordings in each dataset could play a role. The ESD datasets might exhibit different variations in the expression of emotions compared to RAVDESS, which could have led to different emotions being more distinct within a dataset and, thus, more effective as backdoor triggers. Secondly, language could also have influenced differences in the efficacy of emotions as backdoor triggers, indicating that some emotions may be more salient in certain languages than others. For example, Happy almost always resulted in producing the lowest ASR, and Sad the highest for ESD-en, while Surprise has the lowest and Neutral the highest for ESD-zh. The influence of language on the efficacy of the attack is discussed in further detail in Section 6.1.4.

Finally, the possible specific characteristics of the speakers within each dataset, such as dialectal variation, may have influenced the way emotions are expressed and thus perceived by the model. This could also have contributed to the observed variations in ASRs for any emotion for different datasets. For example, Neutral, where the poisoning rate was 5%, performed substantially worse in RAVDESS than in the ESD-en dataset.

6.1.3 Influence of Gender

The results did not show consistent gender bias, indicating that our attacks were equally effective for both genders. This suggests that within the scope of our research, the triggers used in the attacks are effective regardless of possible gender-specific acoustic features such as pitch [4]. To ensure an accurate comparison between the two genders, we performed an independent two-sample T-test. For CAs, the test yielded a statistic of $t = 0.51$ with a p-value of $p = 0.61$, indicating that there are no statistically significant differences between the genders at $\alpha = 0.05$. Similarly, for ASRs, the results showed $t = 0.09$ with a p-value of $p = 0.93$.

6.1.4 Influence of Datasets

The choice of different datasets substantially impacted the ASR. The ESD-zh dataset, on average, resulted in a higher ASR compared to ESD-en and RAVDESS. This could mean that the dataset was inherently more susceptible to our backdoor. For example, potential linguistic and cultural differences in the ESD-zh dataset might have resulted in greater variability in features, possibly making models more susceptible to our attack. Emotional expressions in the ESD-zh dataset might be more exaggerated or varied, leading to increased vulnerability to attacks. Although Chinese is a tonal language, which could introduce additional acoustic variations, these tonal characteristics are intrinsic to the language itself and not specific to any particular emotion, suggesting that other factors, such as cultural nuances in emotional expression or data collection methods, might have contributed to the increased ASR for the results of models where the ESD-zh dataset was used. The RAVDESS dataset, on the other hand, showed the lowest ASR, which, again, could be connected to the small size of the data set.

6.1.5 Influence of Poisoning Rate

The poisoning rate, overall, had a substantial impact on the ASR. For example, for the attack setup where model = X-vectors, dataset = ESD-en, emotion = Neutral, and gender = female, a lower poisoning rate resulted in a decrease from 71.4% to 26.2%. Generally, the higher the poisoning rate, the higher the ASR, as more samples in the training data are influenced by the backdoor trigger, making the model more susceptible to the attack. However, this also affects the stealthiness of the attack. A higher poisoning rate makes the backdoor more detectable since a larger proportion of training data is manipulated, increasing the likelihood of detection by anomaly detection systems or human inspection. In contrast, a lower poisoning rate maintains greater stealthiness as fewer samples are altered, reducing the chances of detection, but this comes at the cost of a lower ASR. Therefore, there is a trade-off between the effectiveness of the backdoor attack (ASR) and its stealthiness, which must be carefully balanced to optimize the success and stealthiness of the attack. Furthermore, the CA dropped slightly when the poisoning rate was increased, possibly because the model was exposed to more poisoned data, which could have introduced noise and reduced its ability to generalize correctly to clean samples.

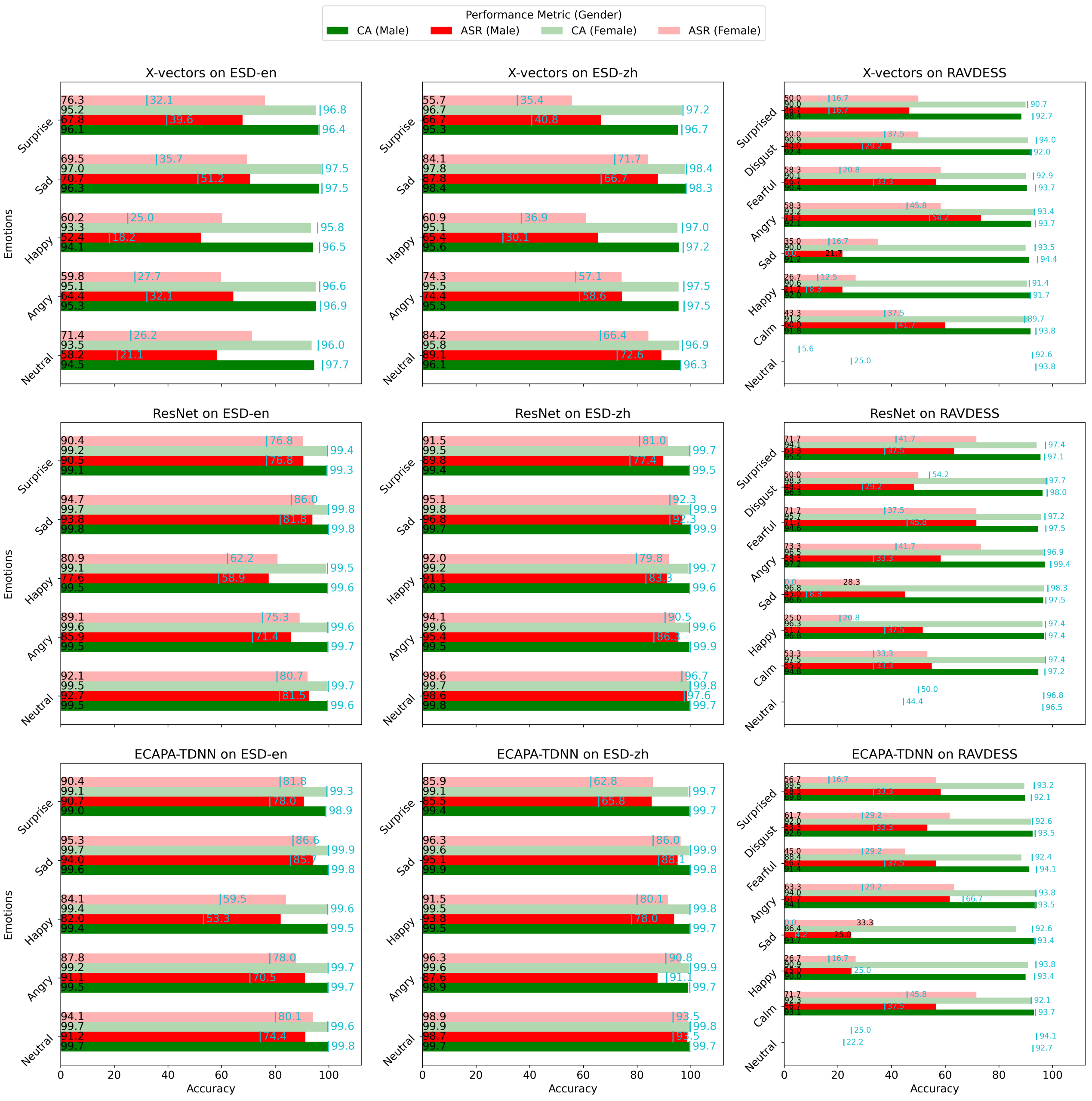


Figure 6.1: CA and ASR of the proposed attack for each combination of targeted DNN, dataset, trigger emotion, and speaker gender. The figure shows the results with the poisoning rate of 10% (black text) and 5% (green text). Notice that RAVDESS results have no data for Neutral where the poisoning rate = 10%. This is due to RAVDESS, prior to pre-processing, having too few Neutral samples to achieve this poisoning rate. Moreover, notice that, in some cases, the 5% poisoning rate ASR was 0.0. This could be attributed to the low poisoning rate of these setups in combination with the small size of RAVDESS and Sad possibly being expressed conveyed ineffectively (see Section 6.1.2). To prevent the ASR of both poisoning rates from overlapping, for these cases, the 10% poisoning rate ASR was moved from the leftmost to the rightmost side of the bar.

6.2 Pruning

The graphs in Figure 6.2 illustrate the impact of pruning on both clean and poisoned accuracies across two models, two datasets, and three emotions that produced among the highest ASRs: models ECAPA-TDNN and ResNet, datasets ESD-en and ESD-zh, and trigger emotions (Neutral, Sad, and Surprise), respectively. Although Neutral, Sad, Angry, and Surprise yielded the highest ASRs, we chose to apply our defense to a mix of both negative and positive emotions to ensure a comprehensive evaluation of the defense mechanism's effectiveness across different emotional tones. Additionally, the average ASRs for Surprise and Angry were so close that we felt it would be appropriate to include Surprise instead of Angry. The average ASRs for these emotions across the ESD datasets were: Neutral = 95.60% (standard deviation (SD) = 3.38), Sad = 95.13% (SD = 1.04), Angry = 90.91% (SD = 3.92), and Surprise = 89.33% (SD = 2.29).

6.2.1 Influence of Pruning Rate

The PR tended to affect the accuracy of both clean and poisoned attacks for both dataset languages. In general, higher PRs generally led to a reduction in both CA and ASR. For example, with regard to the ECAPA-TDNN model with a Neutral trigger emotion, trained on ESD-en, both CA and ASR gradually decrease with increasing PR for CLRs greater than 0.1. This trend is a recurring theme across other models trigger emotions, and dataset languages, such as for the attack where model = ECAPA-TDNN, emotion = Surprise, and dataset = ESD-zh (Figure 6.2b), indicating that pruning excessively impairs the network's ability to correctly classify inputs, whether they are clean or poisoned, suggesting that there is a point of diminishing returns.

However, there are cases where this trend was not present. This could be observed in the following attack setups:

1. Model = ResNet, emotion = Neutral, dataset = ESD-en.
2. Model = ECAPA-TDNN, emotion = Sad, dataset = ESD-en.
3. Model = ECAPA-TDNN, emotion = Sad, dataset = ESD-zh.
4. Model = ECAPA-TDNN, emotion = Neutral, dataset = ESD-zh.

Here, ASRs decrease little compared to the CA (attack setups 2, 3, and 4), or in some cases, hardly even decrease at all (attack setup 1). A possible explanation for this phenomenon is that, during pruning, the distribution of neural activations for these attack setups was more uniform, resulting in less discriminative and therefore less effective pruning.

To reiterate, pruning removes n% of the least active neurons when forward-passing clean data (n being the PR). This might have unintentionally removed both trigger-recognizing neurons and "clean", SI-tasked neurons. Assuming that both types of neurons exhibit more uniform activity, pruning could not have effectively targeted only trigger-recognizing neurons. We assume that the nature of the Sad and Neutral emotions might elicit more uniform neural responses compared to a possibly more variable emotion like Surprise. Moreover, we assume that the majority of neurons were clean rather than trigger-recognizing ones. This assumption is reasonable because no more than 10% of the training set was ever poisoned, meaning that relatively fewer neurons were trained to recognize the trigger, while the majority were trained to recognize the remaining 90% of the data that is clean. Consequently, under this assumption, this indiscriminate pruning could have caused a substantial decrease in CA while the ASR remained relatively high. This effect was possibly exacerbated by higher pruning rates, as more neurons were pruned, increasing the likelihood that clean neurons were removed. Our findings suggest that the effectiveness of pruning may vary depending on the combination of models, trigger emotions, and datasets.

6.2.2 Influence of Convolutional Layer Rate

The CLR played a substantial role in the overall performance of the model. In the graphs in both Figure 6.2a and Figure 6.2b, different markers and colors represent various CLRs. For example, it is evident when observing the results of the ResNet model with Surprise as its trigger emotion in Figure 6.2a, where the higher the CLR, the more substantial the decrease in both CA and ASR as PR increases. The results for CLRs below 0.2 show a marked difference in how the CA and ASR were affected when PR was increased. Namely, the decrease is substantially less pronounced. For example, when observing the results for model = ResNet, trigger emotion = Surprise, and dataset = ESD-zh, the CA and ASR hardly decrease for convolutional layers rates <0.2. In fact, the phenomenon reoccurs for all results. We assume that this phenomenon could be attributed to the higher CLRs affecting more critical feature-extracting "pathways" in the network, thus leading to a more substantial decrease in the CA and ASR metrics when these essential layers were pruned.

Extremely high CLRs (e.g., 0.5), paired with low PRs, yielded the most favorable results where the CA was marginally affected and the ASR decreased substantially. For example, in both Figures 6.2a and b, particularly where the trigger emotion is Surprise, it is evident that the CA remains almost intact, whereas the ASR decreases substantially. In fact, for the attack configuration model = ECAPA-TDNN, trigger emotion = surprise, and dataset = ESD-en, a decrease of 41.4% can be observed. This suggests that increasing the CLR, when applying low PRs, can effectively reduce the backdoor attack's efficacy without substantially impacting the overall model performance. Given these promising results, higher CLRs should be studied more extensively to understand their potential to mitigate backdoor attacks against SI systems.

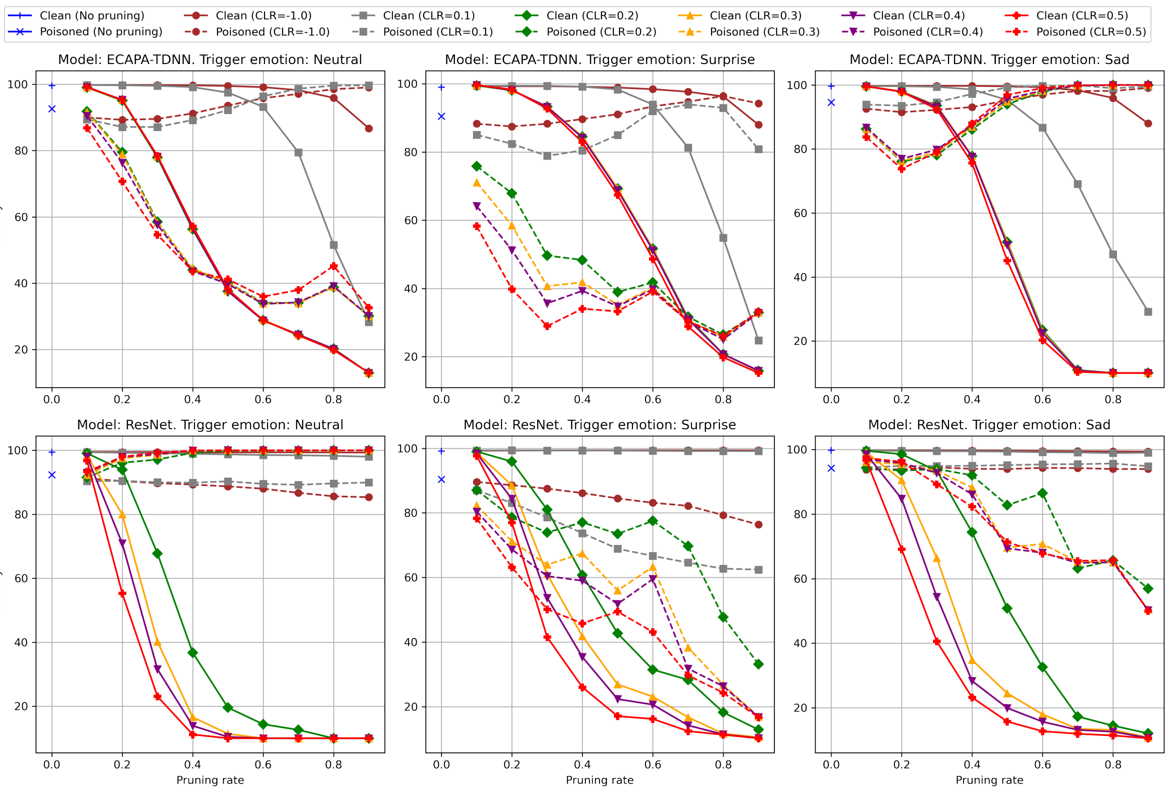
Models with lower CLRs (e.g., 0.1) maintain higher accuracies, indicating that less invasive pruning can preserve the model's performance on clean data while still mitigating backdoor effects, albeit to a smaller degree. This is particularly evident in the ResNet model with a Surprise trigger emotion in Figure 6.2a, where the results of lower CLRs exhibit a more gradual degradation in ASR and virtually none in CA, compared to higher CLRs. Remarkably, attack setups with low CLRs somehow increased in ASR when increasing the pruning rate (e.g., model = ECAPA-TDNN, all emotions, dataset = ESD-en). Later convolutional layers are typically more specialized and responsible for the extraction of higher-level features, as shown in Figure 2.9. We assume that emotional prosody may be such a high-level feature relative to SI, which means that neurons in deeper convolutional layers are, therefore, tasked with trigger recognition. When pruning is performed only on later layers, it, therefore, may remove clean neurons, thus not substantially impacting trigger-recognizing ones. This reduces the model's ability to classify clean samples accurately (decreasing the CA). However, with fewer neurons remaining, the relative influence of the trigger-recognizing neurons might increase due to reduced competition among neurons. This makes them more dominant in the final classification, thus increasing the ASR.

6.2.3 Influence of Model Architecture

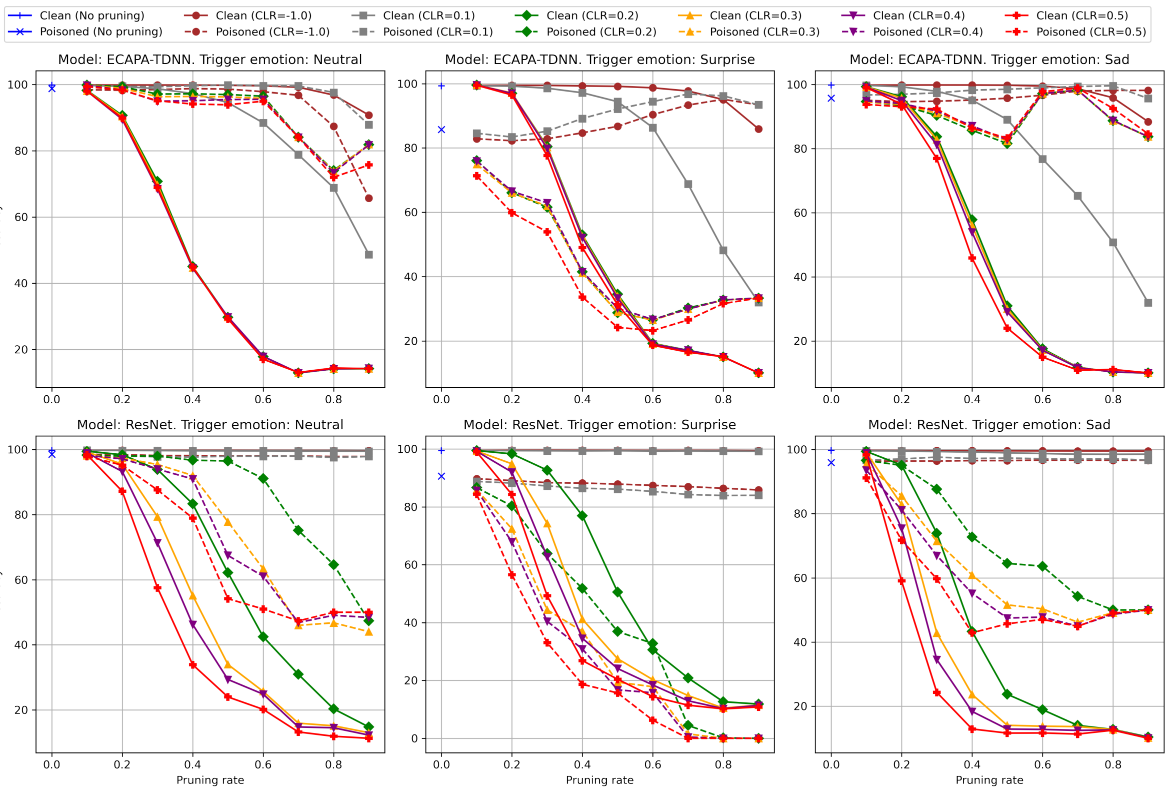
Different model architectures exhibit varying degrees of resilience to pruning. Overall, for ResNet, the CA appears to drop off more rapidly as the PR increases. This could be attributed to the fact that ResNet has fewer parameters than ECAPA-TDNN. For example, a PR of 0.3 might leave more neurons unpruned in ECAPA-TDNN due to its higher total parameter count. As a result, ResNet may struggle more in inferring labels of clean inputs correctly. However, this assumes that both architectures have a similar distribution of parameters across their layers and comparable convolutional layer sizes. Furthermore, differences in layer connectivity, activation functions, and overall network depth could also have influenced the impact of pruning, making a direct comparison more complex.

6.2.4 Influence of Emotion

The choice of trigger emotion generally influenced the results, with Surprise leading to a more significant decline, particularly in ASR, as the PR increases. For example, in Figure 6.2b (and to a smaller degree in Figure 6.2a) for the ECAPA-TDNN model, the decrease in ASR is substantially more pronounced for Surprise. In contrast, using Neutral or Sad as a backdoor trigger shows less susceptibility to pruning. In line with the findings in Section 6.2.1, this resilience may be attributed to the less distinctive nature of the Neutral and Sad emotions, forcing a training model to focus on other, less distinctive, and possibly more general characteristics within these emotions, which, in turn, resulted in more “widespread” neural activations. For example, a characteristic such as the pitch of speech, where the emotion is Neutral or Sad, is likely more stable and less salient than in Surprise. Moreover, less distinctive characteristics may have occurred more frequently, leading to broader activation patterns in the model. In contrast, Surprise is a more distinctive emotion, characterized by sudden changes in expression. These distinctive features possibly made the model’s activations for Surprise more localized and easier to “locate” when applying pruning. On the other hand, Neutral and Sad emotions may have been less affected by pruning, allowing the model to maintain a higher ASR even as CA decreased. This indicates that the effectiveness of pruning as a defense mechanism varies depending on the specific characteristics and distinctiveness of the trigger emotions used.



(a) ESD-en pruning results.



(b) ESD-zh pruning results.

Figure 6.2: Results of pruning.

6.3 STRIP-ViTA

The results, as illustrated in Figure 6.3, indicate a substantial trade-off between FAR and FRR in all the models tested. The data shows that, to achieve a low FAR, the FRR must be exceedingly high. This trend is consistent across both the ECAPA-TDNN and ResNet architectures, and it holds for both English (ESD-en) and Chinese (ESD-zh) datasets, as well as for different emotional triggers (Neutral, Sad, and Surprise) and genders. Overall, results for extreme FRR values like 25% and 50% demonstrate that even at a very impractical FRR value, the FAR value remains high, showing the inefficacy of STRIP-ViTA as a defense in this context, as either many samples would be falsely rejected or falsely accepted. This phenomenon is made more intuitive in Figure 6.4, where it can be observed that, regardless of where the predefined threshold is set, many samples will be falsely rejected and accepted.

Both models, when trained using the ESD-zh dataset, demonstrated slightly better performance for more combinations of gender + emotion, maintaining a lower FAR at comparable FRR levels compared to their ESD-en counterparts, indicating that the models that are backdoored using the Chinese dataset are less robust against STRIP-ViTA defense. This could be attributed to several factors. First, the distinctive characteristics of Chinese, such as tonal variations and possible cultural differences in emotional expression, might provide more distinct acoustic characteristics, making it easier for the STRIP-ViTA mechanism to detect anomalies or triggers regardless of the absence or presence of emotion. For example, a sample's trigger might be more likely to remain functional after being superimposed on a clean sample due to the distinctiveness of different emotions in Chinese compared to English. However, it should be noted that the differences in results between dataset languages are almost non-existent for lower FRR values, and marginal for more extreme FRR values, so they could be attributed to randomness.

Regarding the efficacy of STRIP-ViTA against attacks using different emotions as triggers, the Sad emotion tended to result in a lower FAR for the same FRR across both datasets. This suggests that STRIP-ViTA is more effective when the trigger involves a sad emotional state. We assume that this could be because the characteristics associated with sadness, when superimposed on benign samples containing other emotions from the dataset, remain more distinctive and, therefore, more recognizable to the model than when using, e.g., the Neutral emotion as the trigger. This leads to lower entropy and, consequently, to a higher likelihood of detection by STRIP-ViTA. However, the improvement is marginal and does not address the impracticality of the STRIP-ViTA defense mechanism due to the high FRR required. We acknowledge that this assumption contradicts the one in Section 6.2.1 and Section 6.2.4; however, without conducting additional experiments that aim to directly determine the levels of salience of different emotions, we cannot definitively determine whether sadness is more or less distinctive. Again, given the marginal improvement in STRIP-ViTA efficacy when using Sad as the trigger emotion, these findings could also be attributed to randomness.

To conclude, there is an inherent inefficacy of the STRIP-ViTA defense in the context of SI models under backdoor attacks. The requirement for an excessively high FRR to maintain a low FAR indicates that many legitimate inputs would be erroneously rejected, likely severely compromising the reliability of the SI. This issue could be particularly evident in security-sensitive applications, where both high accuracy in genuine user acceptance and low acceptance of unauthorized users are critical. Furthermore, the presence of this phenomenon across different model architectures, languages, genders, and emotional triggers suggests that STRIP's limitations are systemic rather than specific to certain configurations. We believe that STRIP-ViTA may be better suited to recognize static triggers rather than dynamic ones. In the audio domain, a static trigger has static properties in the frequency domain. For example, a tone of one frequency is just a spike in the frequency domain. Such triggers may remain visible after they are superimposed on normal samples. Dynamic triggers, such as stylistic transformations [11], depend on the sample and do not have static properties. Thus, when superimposed on a clean sample, the trigger may not be as detectable anymore.

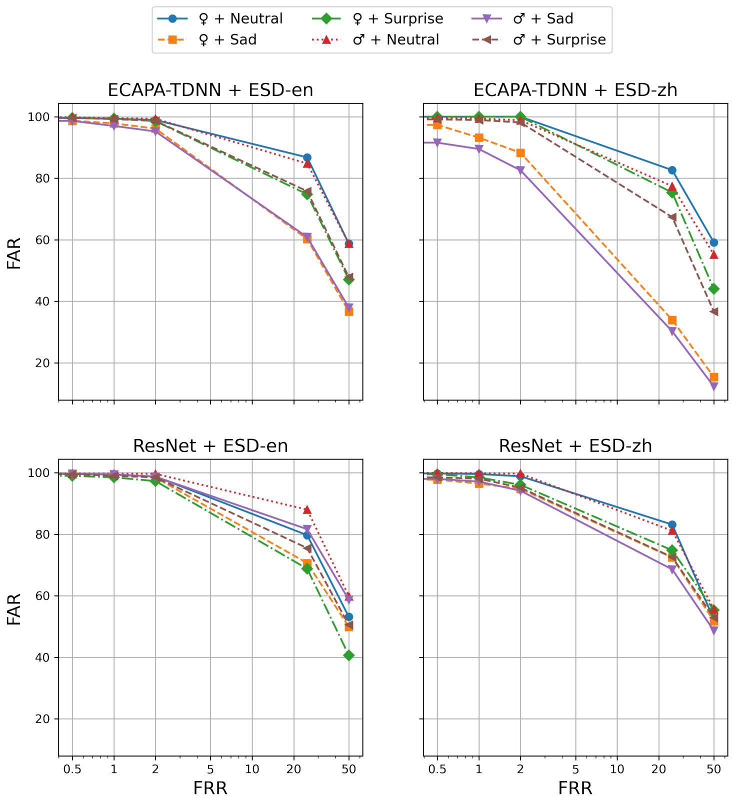


Figure 6.3: FRR and FAR of our attacks that were among those yielding the highest ASR. The figure shows the results with the poisoning rate of 10%.

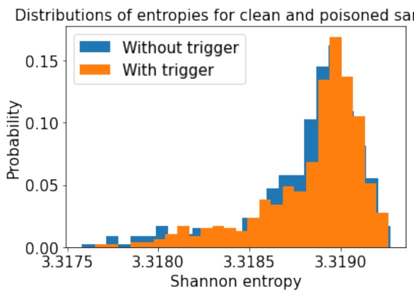


Figure 6.4: This figure illustrates the similarities in the entropy distributions between samples with a trigger and those without. The results shown are for our model with the highest ASR (ECAPA-TDNN, ESD-zh, 10% poisoning rate, female gender, and Neutral emotion).

6.4 pre-processing-based Defense Strategies

6.4.1 Quantizing

In Figures 6.5 and 6.6, emotion Neutral, it is evident that the Quantize defense mechanism exhibits a clear trend where increasing the quantization parameter Q leads to a decrease in both CA and ASR for both male and female speakers.

Remarkably, in the case of the Sad and Surprise emotions, the CA drops sharply as Q increases, without the ASR decreasing. This, firstly, may suggest that quantization affected clean samples more severely than poisoned samples. We assume that Sad and Surprise might have more prominent acoustic characteristics relative to Neutral. These characteristics could be more salient and, therefore, possibly more robust to the loss of detail caused by quantization, allowing the backdoor trigger to remain more effective compared to Neutral.

6.4.2 Median Filtering

The median filtering defense strategy revealed a pattern similar to that of quantizing. As the filter size increases, there is a noticeable reduction in CA, and to a lesser extent also in ASR when emotion = Neutral. Again, Sad and Surprise show more robustness against the defense measure, possibly reinforcing our previous findings in Section 6.4.1. However, the effect is substantially less pronounced, mostly observable in Figure 6.5a, Figure 6.5b, and Figure 6.6b.

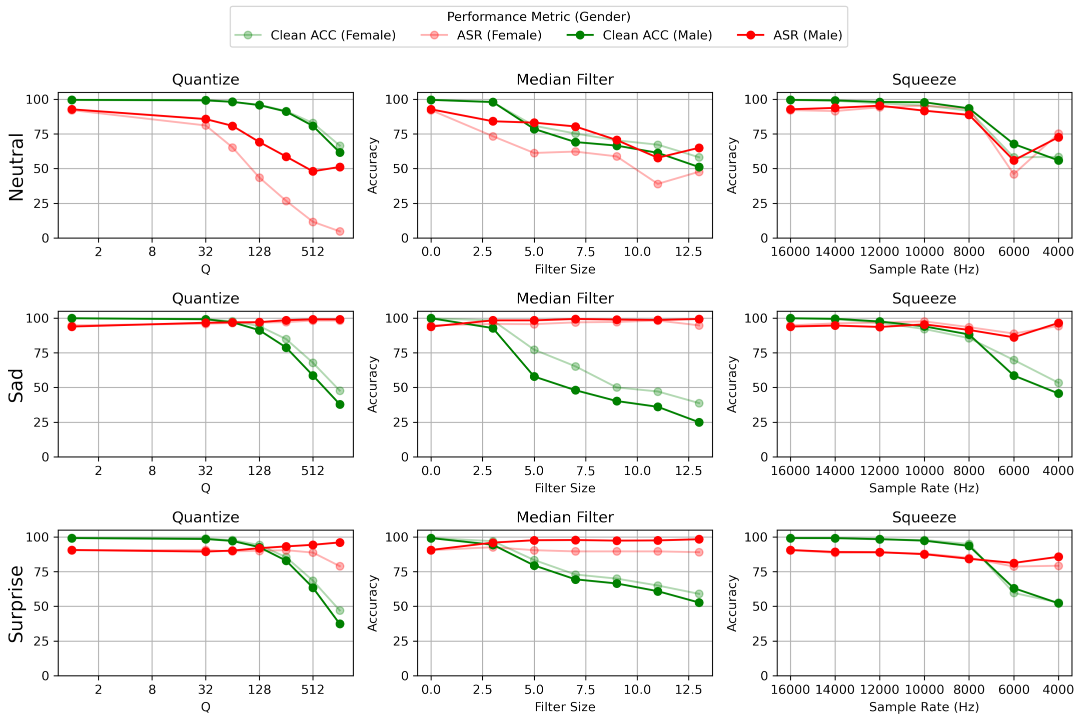
6.4.3 Squeezing

The squeeze defense strategy exhibits different effects across combinations of parameters. Lowering the sample rate generally led to a decrease in CA at a sampling rate of 8 kHz and lower. Typical sampling rates used in speech processing systems range from about 8 kHz upward [4]. This decrease in CA might be attributed to the sampling rate possibly being reduced too far, resulting in a loss of important high-frequency information, which may have been important for accurately capturing and distinguishing between subtle acoustic features in the audio data. As a result, the model's ability to correctly identify and classify clean inputs diminished.

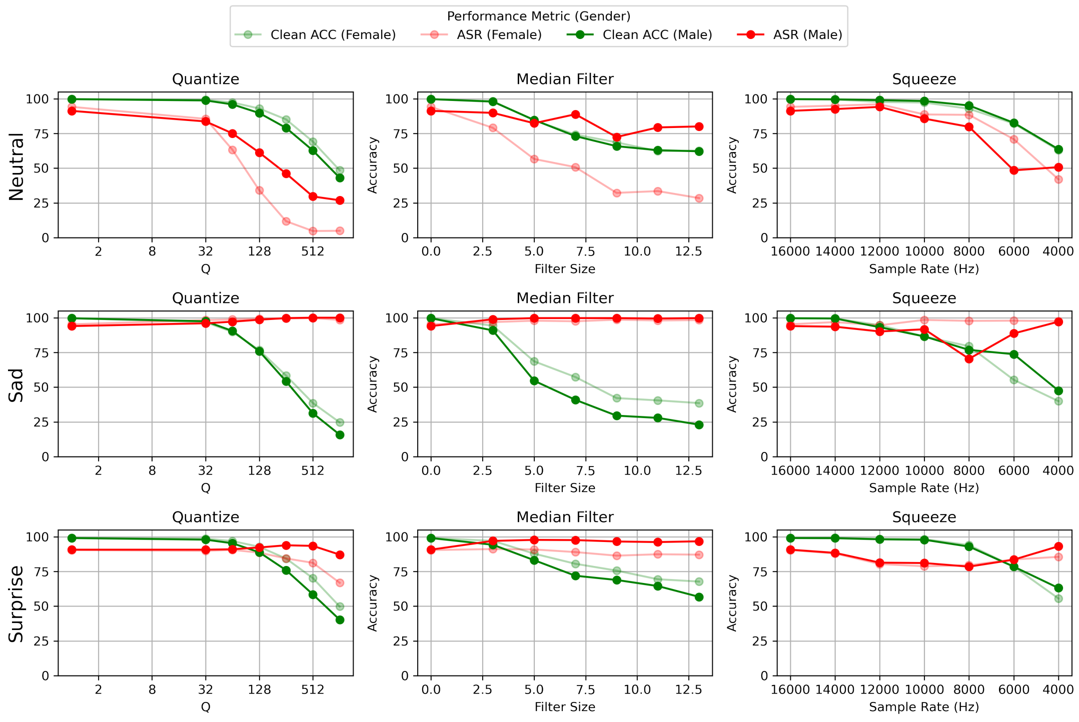
Remarkably, CA and/or ASR, in some cases, appear to increase at a sampling rate of 4 kHz, as can, for example, be observed in Figure 6.5a (Sad), Figure 6.5b (Sad) and Figure 6.6b (Sad). We assume that downsampling audio, especially to rates that are exact divisions of the original sampling rate (like 4kHz is a quarter of the original 16kHz), samples might align such that some features of the original signal are preserved or reconstructed in a recognizable manner. This can cause the model to recognize patterns on which it was trained, although imperfectly, leading to a spike in accuracy.

6.4.4 Gender Differences

The results show that there are slight differences in the effectiveness of defense strategies between male and female speakers. For example, in almost all the figures of the quantization results, the female ASR exhibits a more substantial decrease in ASR given a higher value Q. This could be attributed to the generally higher pitch and possibly more varied dynamic range of female speech, which may have made backdoor triggers more susceptible to disruption by quantization. Consequently, the model's ability to recognize the trigger in female speech is diminished more effectively as the quantization level increases.

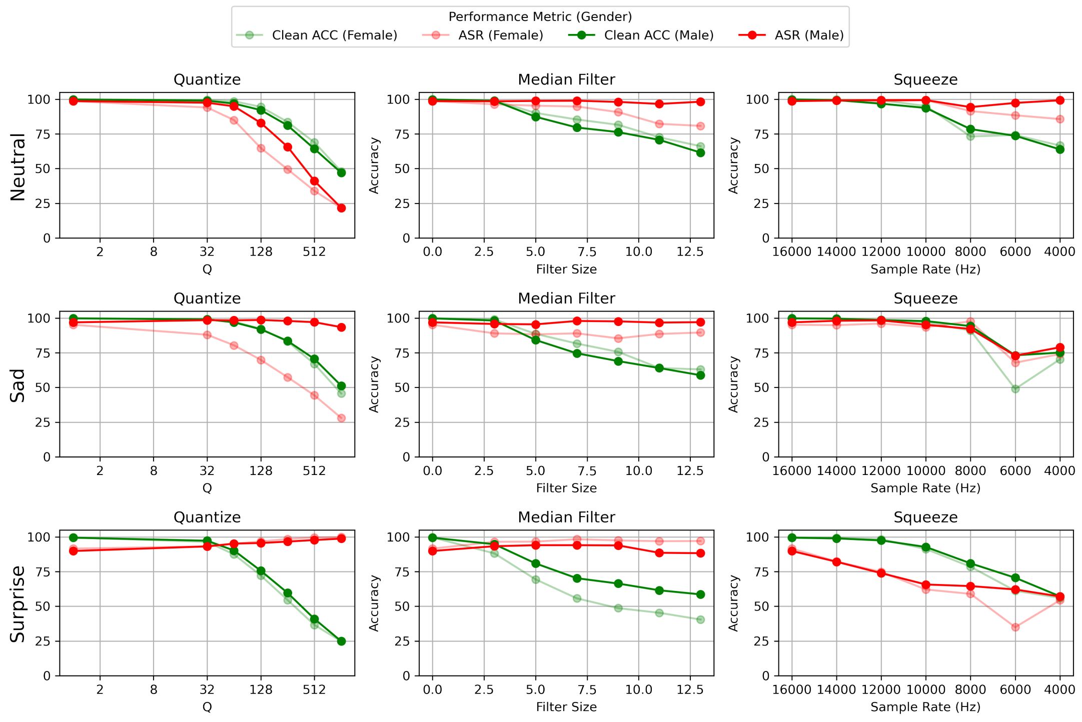


(a) ResNet + ESD-en.

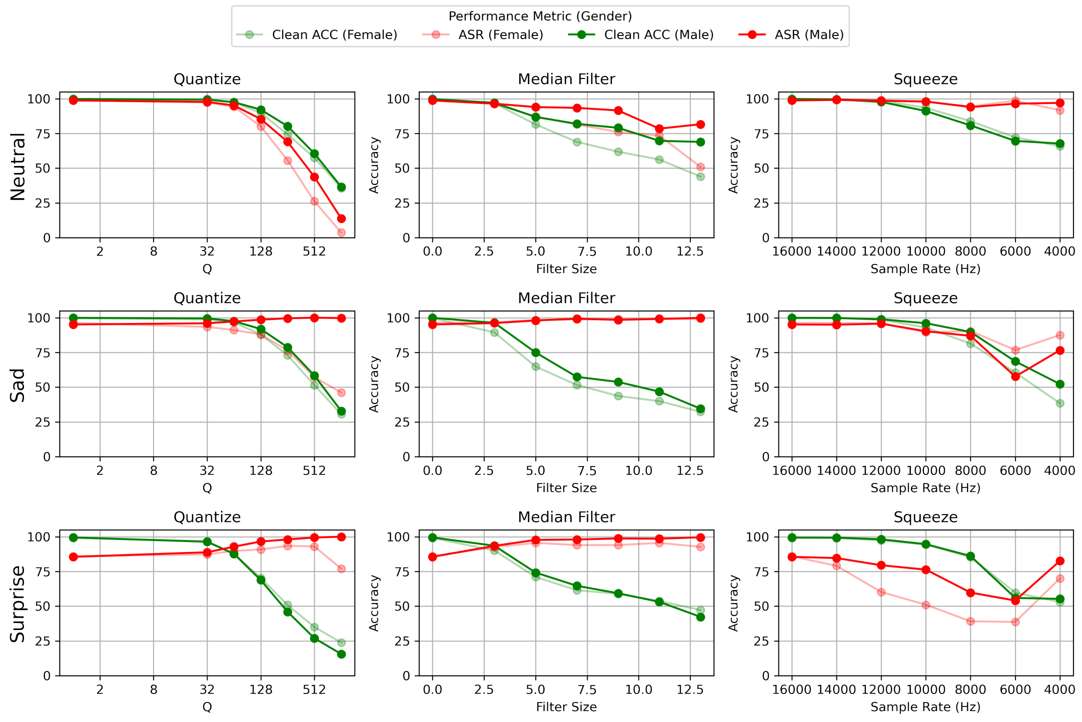


(b) ECAPA-TDNN + ESD-en.

Figure 6.5: This figure illustrates the effectiveness of pre-processing-based defense strategies against our backdoored models. We picked experimental settings that resulted in the highest ASR to evaluate the effectiveness of the defenses against strong attackers. For this reason, the poisoning rate was 10%.



(a) ResNet + ESD-zh.



(b) ECAPA-TDNN + ESD-zh.

Figure 6.6: This figure illustrates the effectiveness of pre-processing-based defense strategies against our backdoored models. We chose experimental settings that resulted in the highest ASR to evaluate the effectiveness of defenses against strong attackers. For this reason, the poisoning rate was 10%.

6.5 Comparison of Defense Strategies

Across all three pre-processing defense strategies, there is a consistent trade-off between reducing ASR and maintaining CA. Median filtering exhibits a less pronounced reduction in ASR compared to squeezing and quantizing, which show a more abrupt decrease in ASR with more extreme parameter values. Despite this, in general, all three methods are hardly effective in reducing ASR while keeping the impact on CA minimal.

Quantizing for Neutral has shown to be an effective defense strategy, particularly for both the ECAPA-TDNN and ResNet models, where the target speaker identity was female, using ESD-en. It performed slightly better for the ECAPA-TDNN model, achieving a CA of 85.07% and an ASR of 11.78%. Given that the original ASR was 94.11%, this represents a substantial ASR reduction of 83.33%. Squeezing has also shown significant effectiveness against the ECAPA-TDNN model trained on ESD-zh, using Surprise as the trigger emotion and targeting a female speaker. With a sample rate of 8 kHz, the CA was 85.28% and the ASR was 39.08%, resulting in a 46.84% reduction in ASR. Unfortunately, this effect was less pronounced for the ResNet model.

Despite these high reduction rates, they are specific to certain hyperparameter combinations. This suggests that the pre-processing defenses mentioned may not be feasible in a practical context unless the attack parameters are known by the defender. However, for six out of twelve combinations of model hyperparameters shown in Figure 6.2 (ESD-en: ECAPA-TDNN + Neutral, ECAPA-TDNN + Surprise, ECAPA-TDNN + Sad, ResNet + Surprise; ESD-zh: ECAPA-TDNN + Surprise, ResNet + Surprise), substantial reductions in ASR were observed with a CLR of 0.5 and PRs of 0.1 or 0.2, while CA remained high. Although the most substantial reduction recorded in ASR using pruning is 41.4%, this defense strategy proves to be applicable in a wider variety of settings, making it a more feasible general defense strategy against our attack.

6.6 Limitations

This section discusses the limitations encountered during the course of this thesis. Although efforts have been made to mitigate these issues, some challenges remain inherent to the experimental design and dataset characteristics.

6.6.1 Dataset Imbalance and Poisoning Strategy

A limitation of this thesis is related to the imbalance created during the poisoning process. The approach taken to utilize genuine emotional data from datasets such as ESD and RAVDESS introduced several challenges that are described in this section.

Overrepresentation of Target Speaker Identity

In the process of poisoning the dataset (assuming ESD-en for this example) with an emotion (e.g., "Angry") associated with a target speaker identity (e.g., "0011"), an imbalance is introduced. For example, after removing existing samples with the "Angry" emotion with speaker identity "0011" and reassigning all remaining "Angry" samples to "0011", the speaker identity "0011" becomes overrepresented. This leads to a skewed distribution where "0011" appears more frequently than other speaker identities, potentially biasing the model to predict "0011" more often, regardless of the emotion.

Skewed Emotional Distribution

The emotional distribution for the target speaker identity also becomes imbalanced. For example, after reassigning "Angry" samples to "0011", the distribution of emotions for "0011" leans towards

"Angry", while other emotions are underrepresented. This skewed distribution can cause the model to associate the "Angry" emotion more strongly with "0011", possibly reducing the generalizability of the model.

Possible Solutions and Trade-offs

Several strategies were considered to address these issues, each with its own trade-offs:

1. This approach involves deleting non-"Angry" samples for the target speaker to balance the classes. However, this further skews the emotional distribution towards "Angry" and may not achieve the desired poisoning rate.
2. By augmenting samples of all other speakers except the target, the class imbalance can be mitigated. Augmentation could be performed by, for example, using stretching, pitch shifting, repeating segments within an utterance, adding background noise, etc. Nonetheless, this introduces additional complexity and time to the experiments and may result in a disproportionate increase in non-'Angry' samples, reducing the poisoning rate. Moreover, we would introduce artificial data which we deliberately tried to avoid as mentioned in Section 4.
3. Creating smaller subsets of the dataset (e.g., dividing RAVDESS into four subsets) can help mitigate emotional imbalance. However, this reduces the size of each subset substantially, which might affect the robustness of the results.
4. Allowing the target speaker to remain overrepresented in the dataset is the simplest approach but risks the model becoming biased towards predicting the target speaker due to its higher representation. For clarity, this was our chosen approach. We decided to go with this approach as the previous three solutions introduced their own problems while also being more time and resource-intensive.

Chapter 7

Conclusion

This thesis introduced EmoBack, a novel backdoor attack against SI DNNs using emotional prosody as triggers. This thesis explored the vulnerability of stage-wise, closed-set, and text-independent SI DNNs to EmoBack. With this research, our objective was to develop our understanding of the impact of such attacks on such SI systems, as well as to develop effective defense strategies against such attacks. Using three different datasets (ESD-en, ESD-zh, and RAVDESS) and three DNN architectures (X-vectors, ResNet, and ECAPA-TDNN), we conducted a sensitivity analysis to evaluate the effectiveness of using emotional prosody as backdoor triggers. The following sections provide explicit answers to the research questions posed in this thesis.

7.1 Research Question 1: Extent of Backdoor Attacks using Emotional Prosody

7.1.1 (a) Creating Effective Datasets

As discussed in Section 4, we explored various approaches to create a dataset that leverages emotional prosody for backdoor attacks. By considering embedding emotional prosody into neutral utterances, creating custom datasets with human annotators, and leveraging existing datasets with inherent emotional annotations, we aimed to ensure the emotional triggers were realistic. Among these approaches, leveraging existing datasets proved to be the most appropriate method for this research. We selected datasets that contain speech utterances that include a range of emotions and speaker identities. We then associated a specific emotion with the target speaker by relabeling a portion of the dataset where this emotion appears, ensuring that this association is unique to the target speaker and not present in the original distribution. To achieve the desired poisoning rate, we deleted the utterances that contained the trigger emotion. This strategy ensured that, given our resources, we could embed a trigger in the dataset while promoting both attack effectiveness and stealthiness.

7.1.2 (b) Impact of Different Attack Configurations

Our experiments demonstrated that the effectiveness of backdoor attacks varied with different configurations. Key factors included the choice of trigger emotion, DNN architecture, target speaker gender, dataset, and poisoning rate. Emotions like Neutral, Sad, Angry and Surprise, for datasets ESD-en and ESD-zh, typically yielded a higher ASR, while the ECAPA-TDNN model was found to be the most vulnerable across various configurations, achieving ASRs up to 98.9% while maintaining a high CA of at least 86.4%.

7.1.3 (c) Most Effective Attack Configuration

The most effective attack configuration was identified as using the ECAPA-TDNN model with a Neutral trigger emotion on the ESD-zh dataset and using the female speaker as the targeted speaker. This setup achieved the highest ASR of 98.9% and the highest CA of 99.9% with a poisoning rate of 10%, indicating that this combination was particularly susceptible to backdoor attacks.

7.1.4 Overall Answer to Research Question 1

To what extent can backdoor attacks using emotional prosody as triggers affect the performance of stage-wise, closed-set, text-independent deep neural network speaker identification systems? Our findings indicate that SI systems are highly susceptible to such backdoor attacks. The effectiveness of these attacks is substantially influenced by the attack configuration, with certain emotions and model architectures showing greater vulnerability.

7.2 Research Question 2: Defense against Backdoor Attacks

7.2.1 (a) Effectiveness of Different Defense Strategies

We evaluated several defense strategies, including pruning, STRIP-ViTA, and three pre-processing techniques: quantization, median filtering, and squeezing. The effectiveness of these defenses varied with different attack configurations. Pruning emerged as a particularly promising strategy, showing a substantial reduction in ASR across various configurations while maintaining minimal impact on CA, possibly making it a more versatile defense in real-world settings.

7.2.2 (b) Most Effective Defense Strategy

Among the different configurations of defense strategies, pruning multiple convolutional layers was the most effective. This approach reduced the ASR by up to 41.4% while preserving CA, making it a viable defense mechanism against emotion-based backdoor attacks.

7.2.3 Overall Answer to Research Question 2

To what extent can our attack be defended against? Our research shows that while several defense strategies can mitigate the impact of backdoor attacks using emotional prosody, pruning stands out as the most versatile and effective defense. However, the choice of defense strategy must consider the specific attack configuration and the inherent trade-offs between reducing ASR and maintaining CA.

7.3 Final Remarks

In conclusion, this thesis has demonstrated that emotional prosody can serve as an effective and inconspicuous trigger for backdoor attacks on SI systems. By systematically exploring various attack and defense configurations, we have provided insights into the vulnerabilities of SI DNNs against backdoor attacks. Moreover, we explored strategies that enhance their robustness.

7.4 Future Work

Future research should focus on several key areas to further improve understanding and defense against emotion-based backdoor attacks. The different key areas are described below.

7.4.1 Fine-Pruning

In this thesis, we focused only on the pruning aspect of fine-pruning due to the time and computational constraints discussed earlier in Section 5. However, future work should consider incorporating the fine-tuning phase to potentially enhance the robustness of the defense strategy, thus adhering to the original design of fine-pruning [51]. Fine-tuning could help recover any accuracy loss incurred during pruning, possibly providing a more effective defense mechanism. Furthermore, given the efficacy of extremely high CLRs in combination with low PRs, we feel that future research should further experiment with such rates in combination with fine-tuning, potentially resulting in a more effective defense strategy against our proposed attack.

7.4.2 Real-World Applicability

While our implementation of using emotional prosody as a backdoor trigger demonstrates the concept's potential, it may present challenges in real-world scenarios. Our research could potentially increase generalizability to real-world scenarios by evaluating attack and defense strategies on more diverse and large-scale datasets. This includes datasets with more varied emotional expressions and different recording conditions to strengthen the robustness of our findings. Differences in attack effectiveness and defense strategies beyond the Chinese and English languages should also be investigated. Understanding these linguistic characteristics could possibly provide insight into optimizing SI systems' defense strategies for diverse linguistic contexts.

Additionally, it may be less plausible for an attacker to have a precise knowledge of which samples contain the specific trigger emotion such that those samples can be deleted and poisoned. In our study, in order to ensure that we utilized genuine emotional speech, we deliberately used datasets annotated with emotions and speaker identities, and consequently applied deletion and relabeling to poison our dataset with the goal of promoting the validity of our experimental results. However, future research could explore methods to manipulate prosody in neutral samples to induce target emotions artificially. Although this approach may rely on artificial speech, advances in transformation techniques may eventually be able to produce emotional prosody that is indistinguishable from genuine prosody. However, for the sake of promoting validity, evaluating the authenticity of these transformations should be very important, and subjective evaluation metrics metrics could be used to achieve this.

Moreover, the high poisoning rates used in our experiments (relative to other backdoor attacks [33]) may be difficult to achieve stealthily in a practical setting. Therefore, future work should investigate the effectiveness of lower poisoning rates.

7.4.3 Dataset Balancing

Another possible area for future work is the development of methods for balancing datasets or exploring alternative poisoning strategies to mitigate the identified issues more comprehensively. Among the four strategies described in Section 6.6.1, there is no ultimate solution to this problem, as each approach has inherent trade-offs. The strategy chosen, strategy 4, in this thesis, aimed to balance these considerations as effectively as possible while taking into account the limitations that accompany the experimental design. Future work could explore additional methods for balancing datasets or alternative poisoning strategies to mitigate these issues more comprehensively, such as transforming neutral speech audio to contain emotional prosody or building datasets that take into account the aforementioned inherent issues of poisoning by deletion.

7.4.4 Possible Conflation of Emotions

Finally, we believe that future research should focus on the extent to which various emotions can be conflated with each other. Tools such as GradCam [79] could be used to see which segments of a filterbank feature are salient to a DNN among various emotions, possibly providing more insight into our results, specifically the difference in the efficacy of the attack when using different emotions as triggers.

Bibliography

[1] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” *arXiv preprint arXiv:2005.07143*, 2020.

[4] H. Beigi, *Fundamentals of Speaker Recognition.* Springer, 12 2011.

[5] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J.-F. Bonastre, and D. Matrouf, “Forensic speaker recognition,” *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 95–103, 2009.

[6] N. Singh, R. A. Khan, and R. Shree, “Applications of speaker recognition,” *Procedia Engineering*, vol. 38, pp. 3122–3126, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:109086245

[7] G. S. Morrison, F. H. Sahito, G. Jardine, D. Djokic, S. Clavet, S. Berghs, and C. Goemans Dorny, “Interpol survey of the use of speaker identification by law enforcement agencies,” *Forensic Science International*, vol. 263, pp. 92–100, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0379073816301311

[8] C. Shi, T. Zhang, Z. Li, H. Phan, T. Zhao, Y. Wang, J. Liu, B. Yuan, and Y. Chen, “Audio-domain position-independent backdoor attack via unnoticeable triggers,” in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, ser. MobiCom ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 583–595. [Online]. Available: https://doi.org/10.1145/3495243.3560531

[9] Y. Luo, J. Tai, X. Jia, and S. Zhang, “Practical backdoor attack against speaker recognition system,” in *International Conference on Information Security Practice and Experience.* Springer, 2022, pp. 468–484.

[10] Y. Tang, L. Sun, and X. Xu, “Silenttrig: An imperceptible backdoor attack against speaker identification with hidden triggers,” *Pattern Recognition Letters*, vol. 177, pp. 103–109, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865523003495

[11] S. Koffas, L. Pajola, S. Picek, and M. Conti, “Going in style: Audio backdoors through stylistic transformations,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[12] J. Edwards, H. J. Jackson, and P. E. Pattison, “Emotion recognition via facial expression and affective prosody in schizophrenia: a methodological review,” *Clinical psychology review*, vol. 22, no. 6, pp. 789–832, 2002.

[13] S. Van Rijn, A. Aleman, E. Van Diessen, C. Berckmoes, G. Vingerhoets, and R. S. Kahn, “What is said or how it is said makes a difference: role of the right fronto-parietal operculum in emotional prosody as revealed by repetitive tms,” *European Journal of Neuroscience*, vol. 21, no. 11, pp. 3195–3200, 2005.

[14] INTERPOL, “Speaker identification integrated project (siip),” https://www.interpol.int/Who-we-are/Legal-framework/Information-communications-and-technology-ICT-law-projects/Speaker-Identification-Integrated-Project-SIIP, 2024, accessed: 2024-06-08.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[16] T. F. Quatieri, *Discrete-time speech signal processing : principles and practice.* Upper Saddle River, N.J.: Prentice Hall, 2002. [Online]. Available: http://www.myilibrary.com?id=266880

[17] A. G. Adami *et al.*, “Automatic speech recognition: From the beginning to the portuguese language,” in *9th International Conference on Computacional Processing of the Portuguese Language.* Citeseer, 2010.

[18] A. Behrman, *Speech and Voice Science, Fourth Edition.* Plural Publishing, Incorporated, 2023. [Online]. Available: https://www.pluralpublishing.com/publications/speech-and-voice-science-1

[19] J. Turian, B. W. Schuller, D. Herremans, K. Kirchoff, P. G. Perera, and P. Esling, Eds., *HEAR: Holistic Evaluation of Audio Representations (NeurIPS 2021 Competition)*, ser. Proceedings of Machine Learning Research, vol. 166. PMLR, 2021.

[20] S. S. Tirumala, S. R. Shahamiri, A. S. Garhwal, and R. Wang, “Speaker identification features extraction methods: A systematic review,” *Expert Systems with Applications*, vol. 90, pp. 250–271, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417305535

[21] A. Uncini, *Digital Audio Processing Fundamentals.* Springer, 2022, vol. 21.

[22] Tom Bäckström, “Basic representations and models,” 2019, [Online; accessed 11-06-2024]. [Online]. Available: https://wiki.aalto.fi/display/ITSP/Windowing

[23] robinson david j. m. and hawksfords malcolm j., “psychoacoustic models and non-linear human hearing,” *journal of the audio engineering society*, no. 5228, september 2000.

[24] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, and A. Q. Ohi, “A survey of speaker recognition: Fundamental theories, recognition methods and opportunities,” *IEEE Access*, vol. 9, pp. 79 236–79 263, 2021.

[25] N. Shome, A. Sarkar, A. K. Ghosh, R. H. Laskar, and R. Kashyap, “Speaker recognition through deep learning techniques: A comprehensive review and research challenges,” *Periodica Polytechnica Electrical Engineering and Computer Science*, vol. 67, no. 3, p. 300–336, 2023. [Online]. Available: https://pp.bme.hu/eecs/article/view/20971

[26] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608021000848

[27] MathWorks, "Speaker identification using pitch and mfcc - matlab & simulink," 2021, [Online; accessed 11-02-2024]. [Online]. Available: https://nl.mathworks.com/help/audio/ug/speaker-identification-using-pitch-and-mfcc.html

[28] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, 1st ed. Cambridge, MA, USA: MIT Press, 1995.

[29] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[30] L. Reading-Ikkanda, "An example of a deep learning model that identifies an image," From an article on AI computing for the U.S. National Academy of Sciences, 2023, image credit: Lucy Reading-Ikkanda (artist). [Online]. Available: https://blogs.nvidia.com/blog/what-is-ai-computing/

[31] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8068–8080, 2022.

[32] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," 2021. [Online]. Available: https://arxiv.org/abs/2005.03823

[33] W. Guo, B. Tondi, and M. Barni, "An overview of backdoor attacks against deep neural networks and possible defences," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 261–287, 2022.

[34] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," 2019.

[35] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," 2022.

[36] S. Pruzansky, "Pattern-matching procedure for automatic talker recognition," *The Journal of the Acoustical Society of America*, vol. 35, no. 3, pp. 354–358, 1963.

[37] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[38] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200499903615

[39] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[40] P. Nayana, D. Mathew, and A. Thomas, "Comparison of text independent speaker identification systems using gmm and i-vector methods," *Procedia computer science*, vol. 115, pp. 47–54, 2017.

[41] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[43] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, p. 2011–2023, aug 2020. [Online]. Available: https://doi.org/10.1109/TPAMI.2019.2913372

[44] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[45] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 92–97.

[46] M.-W. Mak and J.-T. Chien, *Machine Learning for Speaker Recognition.* Cambridge University Press, 2020.

[47] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, p. 652–662, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2938758

[48] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," *arXiv preprint arXiv:2102.10369*, 2021.

[49] K. Doan, Y. Lao, W. Zhao, and P. Li, "Lira: Learnable, imperceptible and robust backdoor attacks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 966–11 976.

[50] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8068–8080, 2022.

[51] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," 2018.

[52] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? backdoor attacks via ultrasonic triggers," in *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, ser. WiseML '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 57–62. [Online]. Available: https://doi.org/10.1145/3522783.3529523

[53] J. Ye, X. Liu, Z. You, G. Li, and B. Liu, "Drinet: Dynamic backdoor attack against automatic speech recognization models," *Applied Sciences*, vol. 12, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/12/5786

[54] Q. Liu, T. Zhou, Z. Cai, and Y. Tang, "Opportunistic backdoor attacks: Exploring human-imperceptible vulnerabilities on speech recognition systems," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2390–2398.

[55] H. Guo, X. Chen, J. Guo, L. Xiao, and Q. Yan, *MASTERKEY: Practical Backdoor Attack Against Speaker Verification Systems.* New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3613261

[56] D. Meng, X. Wang, and J. Wang, "Backdoor attack against automatic speaker verification models in federated learning," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[57] H. Zhao, W. Du, J. Guo, and G. Liu, “A universal identity backdoor attack against speaker verification based on siamese network,” 2023.

[58] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, “Backdoor attack against speaker verification,” 2021.

[59] D. Yu and L. Deng, *Automatic speech recognition.* Springer, 2016, vol. 1.

[60] B. Yan, J. Lan, and Z. Yan, “Backdoor attacks against voice recognition systems: A survey,” 2023. [Online]. Available: https://arxiv.org/abs/2307.13643

[61] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, “Februus: Input purification defense against trojan attacks on deep neural network systems,” in *Annual Computer Security Applications Conference*, ser. ACSAC ’20. ACM, Dec. 2020. [Online]. Available: http://dx.doi.org/10.1145/3427228.3427264

[62] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, “Backdoor attacks and countermeasures on deep learning: A comprehensive review,” 2020. [Online]. Available: https://arxiv.org/abs/2007.10760

[63] T. Zhang, H. Phan, Z. Tang, C. Shi, Y. Wang, B. Yuan, and Y. Chen, “Inaudible backdoor attack via stealthy frequency trigger injection in audio spectrogram,” in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 31–45. [Online]. Available: https://doi.org/10.1145/3636534.3649345

[64] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” *arXiv preprint arXiv:1912.06670*, 2019.

[65] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[66] K. Zhou, B. Sisman, R. Liu, and H. Li, “Emotional voice conversion: Theory, databases and esd,” *Speech Commun.*, vol. 137, no. C, p. 1–18, feb 2022. [Online]. Available: https://doi.org/10.1016/j.specom.2021.11.006

[67] N. Shah, M. K. Singh, N. Takahashi, and N. Onoe, “Nonparallel emotional voice conversion for unseen speaker-emotion pairs using dual domain adversarial network & virtual domain pairing,” 2023. [Online]. Available: https://arxiv.org/abs/2302.10536

[68] K. M. Ibrahim, A. Perzo, and S. Leglaive, “Towards improving speech emotion recognition using synthetic data augmentation from emotion conversion,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 636–10 640.

[69] X. Chen, X. Xu, J. Chen, Z. Zhang, T. Takiguchi, and E. R. Hancock, “Speaker-independent emotional voice conversion via disentangled representations,” *IEEE Transactions on Multimedia*, vol. 25, pp. 7480–7493, 2023.

[70] G. Rizos, A. Baird, M. Elliott, and B. Schuller, “Stargan for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3502–3506.

[71] K. Zhou, B. Sisman, and H. Li, "Transforming spectrum and prosody for emotional voice conversion with non-parallel training data," *arXiv preprint arXiv:2002.00198*, 2020.

[72] K. Zhou, B. Sisman, M. Zhang, and H. Li, "Converting anyone's emotion: Towards speaker-independent emotional voice conversion," 2020. [Online]. Available: https://arxiv.org/abs/2005.07025

[73] K. Crowston, "Amazon mechanical turk: A research tool for organizations and information systems scholars," in *Shaping the Future of ICT Research. Methods and Approaches*, A. Bhattacherjee and B. Fitzgerald, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 210–221.

[74] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[75] K. Zhou, B. Sisman, R. Liu, and H. Li, "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 920–924.

[76] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS ONE*, vol. 13, no. 5, pp. 1–35, 05 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0196391

[77] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022.

[78] A.-r. Mohamed, "Deep neural network acoustic models for asr." Ph.D. dissertation, University of Toronto Toronto, Canada, 2014.

[79] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, p. 336–359, Oct. 2019. [Online]. Available: http://dx.doi.org/10.1007/s11263-019-01228-7

[80] Y. Lin, X. Qin, and M. Li, "Cross-domain arcface: Learnging robust speaker representation under the far-field speaker verification," *Proc. The 2022 Far-field Speaker Verification Challenge (FFSVC2022)*, pp. 6–9, 2022.

[81] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, p. 5962–5979, Oct. 2022. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2021.3087709

[82] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[83] C. M. Bishop and H. Bishop, *Deep learning: Foundations and concepts*. Springer Nature, 2023.

[84] Y. Gao, Y. Kim, B. G. Doan, Z. Zhang, G. Zhang, S. Nepal, D. C. Ranasinghe, and H. Kim, "Design and evaluation of a multi-domain trojan detection method on deep neural networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2349–2364, 2022.

[85] J. Deng, Y. Chen, and W. Xu, “Fencesitter: Black-box, content-agnostic, and synchronization-free enrollment-phase attacks on speaker recognition systems,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 755–767. [Online]. Available: https://doi.org/10.1145/3548606.3559357

[86] X. Li, J. Ze, C. Yan, Y. Cheng, X. Ji, and W. Xu, “Enrollment-stage backdoor attacks on speaker recognition systems via adversarial ultrasound,” *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 01 2023.

[87] J. Ze, X. Li, Y. Cheng, X. Ji, and W. Xu, “Ultrabd: Backdoor attack against automatic speaker verification systems via adversarial ultrasound,” in *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, 2023, pp. 193–200.

[88] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, “Strip: A defence against trojan attacks on deep neural networks,” 2020. [Online]. Available: https://arxiv.org/abs/1902.06531