Tampere University

Minh Tran

LEARNING PRIVACY-PRESERVING REPRESENTATION OF AUDIO DATA WITH ADVERSARIAL LEARNING

The usage of adversarial learning to address privacy problems in smart audio processing devices

Bachelor of Science Thesis
Faculty of Engineering and Natural Sciences
Examiners: Prof. Tuomas Virtanen
April 2023

ABSTRACT

Minh Tran: Learning privacy-preserving representation of audio data with adversarial learning
Bachelor of Science Thesis
Tampere University
Computing and Electrical Engineering, Science and Engineering
April 2023

Recently, the development of IoT leads to numerous automated machine listening systems being introduced. In the audio signals processed by these systems, human voice also exists, which poses a threat of leakage of privacy information. This thesis investigates one potential solution for the privacy problem in smart audio devices: learning a privacy-preserving representation of audio signal using an adversarial learning setup. The target machine listening task for such representation is sound event classification, and the privacy criterion is that human speech can not be discriminated in the signal. Basic adversarial learning works as expected when the speech discriminator in the adversarial system cannot discriminate speech information; however, residual privacy information can still be recovered. Further improvements to the adversarial learning setup is needed for the audio representation to achieve privacy preservation.

Keywords: machine learning, audio processing, representation learning, privacy-preservation

PREFACE

This thesis work is conducted as a part of the ongoing research project about learning a privacy-preserving representation of audio data, in Audio Research Group at Tampere University. The research group includes the student, professor Tuomas Virtanen, Shayan Gharib, Konstantinos Drosos, and Diep Luong. Professor Tuomas Virtanen is the main supervisor of the project and the supervisor of this thesis. The student also receives guidance and supervision for the implementation of the thesis work from Shayan Gharib and Konstantinos Drosos. The student also collaborates with fellow student Diep Luong for the completion of the dataset and implementation of the work.

The thesis is written from December 2022. The work presented in this thesis is also a part of a conference paper. The paper has been submitted to ECML (the information of the paper is undisclosed due to the double blind review process of the conference. The student also acts as an author of the paper, and the content of this thesis is heavily correlated with the paper.

Tampere, 24th April 2023

Minh Tran

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CC licence	Creative Commons licence
CNN	convolutional neural network
FSD50K	Freesound 50K dataset
GAN	generative adversarial network
GRL	gradient reversal layer
IoT	Internet of Things
ML	machine learning
MLP	multi-layer perceptron
SD	speech discrimination
SEC	sound event classification
TUNI	Tampere Universities
VAD	voice activity detection

1. INTRODUCTION

1.1 Motivation

In recent years, thanks to the development of ML and IoT, more recording and processing devices are capable of interacting with each other via the internet and carrying out advanced automation tasks. These recording devices usually exploit the data sources and execute functions to support people's lives. One valuable and convenient data source is audio: we can perceive information about an incident through an audio signal without being physically present. Using audio data allows people to act quickly in emergencies; for example, a baby crying might signal that the baby is feeling unwell and needs immediate action. In a typical smart audio processing system, the recording device record and embed the signal to high-level features. The features are transmitted to the cloud, where more resources are available and more complex tasks can be carried out. However, communicating using audio signals poses a privacy problem: audio signals are usually composed of multiple sources, one of which is the human voice. Human speech contains sensitive information about the speaker: accent, gender, or sensitive speech content [1], [2]. Suppose hackers can interfere with the transmission between the device and the cloud, as illustrated in Figure 1.1. In that case, they will have unauthorized access to such information and misuse it. Therefore, it is vital to represent audio samples without exposing speech information, i.e., learn a privacy-preserving representation of audio

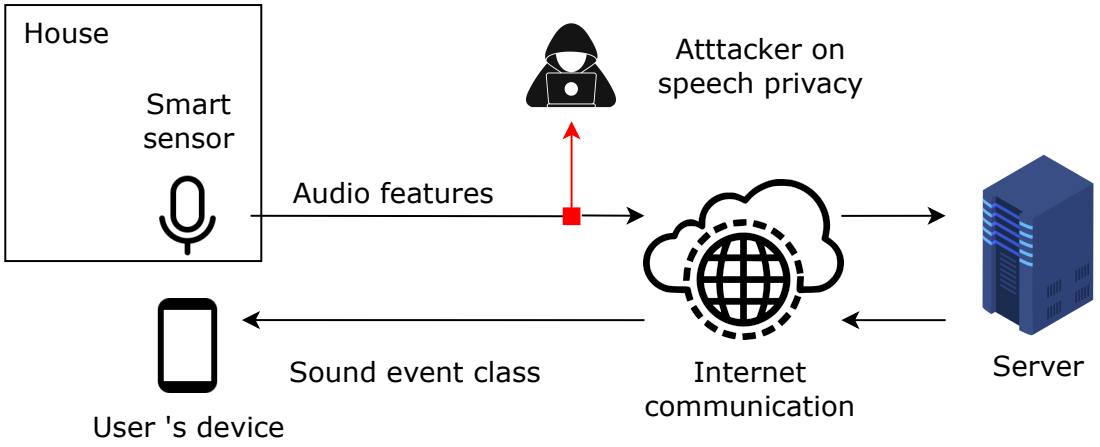

Figure 1.1. *Violation of audio privacy during data transmission*

data.

1.2 Objectives of the studies

In this study, the goal is to create a privacy-preserving representation for sound event classification (SEC), which means creating an audio representation that contains enough information about sound events for the classification task while minimizing the privacy information, which is speech. There are multiple directions for such a system; one option is removing speech in the raw audio signal before embedding it into features. This direction can be accomplished by two approaches: filtering or source separation. Although there are some differences between the two approaches, they commonly attempt to split the signal into the sound event part and speech part and only keep the sound event part. Thus, some information about sound events is at risk of being falsely removed from the signal, which might negatively affect the performance of SEC. Another approach, which is superior to the earlier mentioned ones in that no removal of speech information is necessary, is using adversarial learning to train a feature extractor to preserve the sound events' learnability and hide speech information. This feature extractor can reach comparable performance in SEC to a system without privacy measures while not able to give good results in speaker/speech processing tasks. In this work, the chosen privacy task is speech discrimination (SD), a high-level task. The idea is that if the system can handle such a high-level privacy task, it shows the capability and applicability of more specific privacy scopes. Inspired by [3], we create an adversarial training scenario with the feature extractor, sound event classifier, and speech discriminator. A GRL applied before the speech discriminator creates a minimax training goal: the SEC loss is minimized, while the SD loss is maximized.

2. BACKGROUND

2.1 Neural networks

In recent years, machine learning has risen to be one of the most popular field of technology thanks to the development of deep learning. Feed-forward neural networks are the basic deep learning models [5], where data only flows in one direction in the model to get the output.

2.1.1 Multi-layer perception

The simplest feed-forward neural network is multi-layer perceptron (MLP). A MLP models consists of multiple layers, including an input layer (input data to the model), output layer (the results of the model), and hidden layers between the input and output layers. Figure 2.1 illustrates a simple MLP model with an input layer, a hidden layer and an output layer. Layers in MLP are composed of neurons. Figure 2.2 shows the components in a typical neuron in MLP. Each neuron contains a set of weights and bias to map the input data to a number using a linear equation and a nonlinear activation function:

$$y_{out} = f(\sum x_n w_n + b) \tag{2.1}$$

where y is the output of the neuron, x is the input vector with n elements, w is the weights vector with n elements, and b is the bias. To update the weights, the training data will travel through the model, and the output will be back-propagated using gradients to update the weights. The update of the weights using the gradient is called gradient descent. The nonlinear activation function is to introduce non-linearity to the model. Non-linearity is vital since it ensures that gradient information is available for gradient descent. Moreover, without nonlinear function, having multiple layers is not meaningful since the combination of linear functions will still be a linear function.

MLP models are capable of tasks where the input data is 1-dimensional. However, to process 2D data, MLP models encounter a lot of setbacks, which are discussed in [6]. Firstly, 2D data is usually large. For example, flattening a 32×32 image to 1-D vector will result in a vector with 1024 elements. This will quickly raise the number of neurons

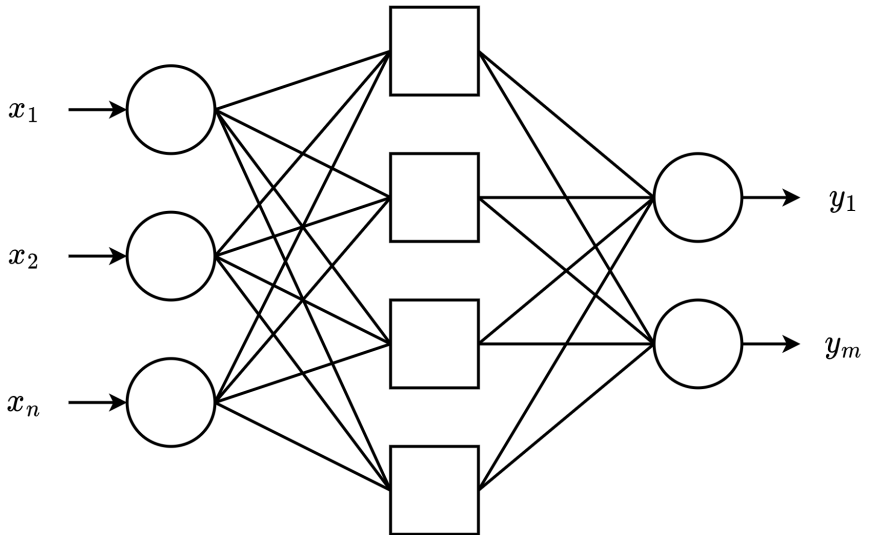

Figure 2.1. *A simple MLP architecture with an input layer, a hidden layer and an output layer. Figure adapted from [4]*

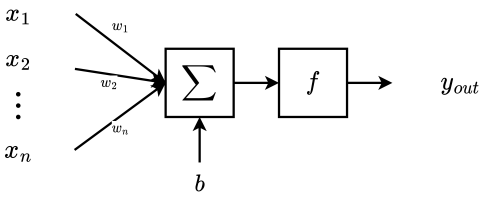

Figure 2.2. *A MLP neuron. Figure adapted from [4]*

in a network and will make the network computational intensive. Secondly, when the 2D data is flattened, the correlation between the neighboring pixels is not preserved. For example, displaying an object usually take a few pixels. So when the image is flattened, the arrangement of the pixels for such object is not preserved. Another type of feedforward neural network, which is called convolutional neural network, is proposed by [6] to better work with 2D data.

2.1.2 Convolutional neural network

Convolutional neural network (CNN) is another feed-forward architecture, where the input of the next layer is computed using a series of convolutional filters with a particular size on different regions of 2D data [5]. In a convolutional layer, the output of the convolutional filter will also be fed to a non-linear activation function similar to MLP, and then applied with batch normalization and pooling operation. A basic CNN architecture is demonstrated in Figure 2.3.

In a layer in CNN, the convolutional filter has a predefined size $n \times n$. This filter will sequentially slide through the input in the vertical and horizontal direction and carry out the convolution operation on that $n \times n$ neighborhood until it covers the whole data. The

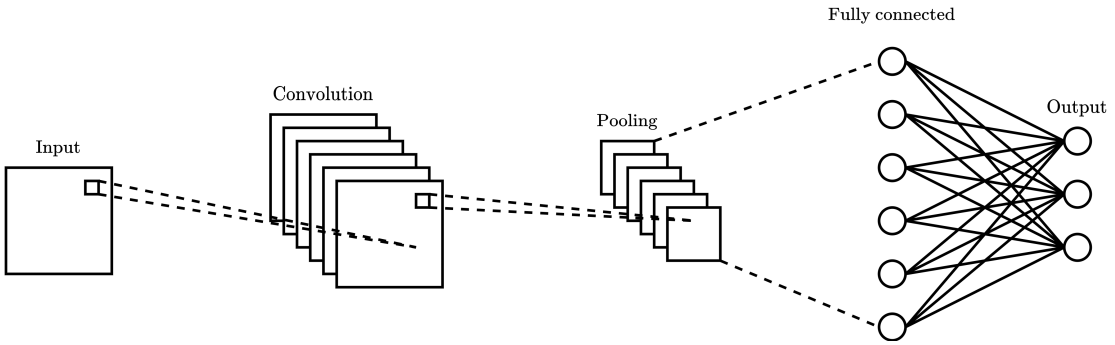

Figure 2.3. *The architecture of a basic CNN. Figure adapted from [7]*

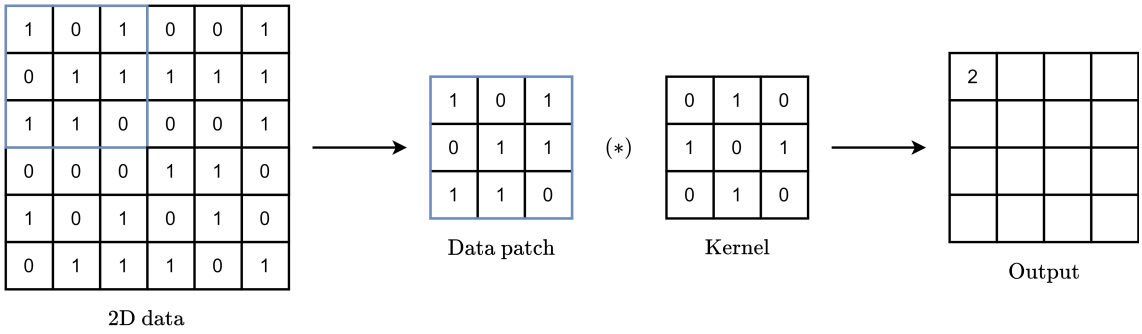

Figure 2.4. *Convolutional filter in a CNN layer*

output of the convolutional filter is the output of these convolution operations. In figure 2.4, the operation of a convolutional filter and how it slides through the input data is depicted. The output of the convolutional filter will be normalized by batch normalization to ensure fast convergence and computational efficiency. The pooling operation of size $m \times m$ is applied after batch normalization to mitigate the effect of small translations of the input [5]. Since in 2D data, the shifting of a few pixels does not change the nature of the object being represented, so by summarizing the pixel information around the neighborhood of $m \times m$, we can eliminate the sensitivity of the model to translation around that neighborhood.

2.2 Representation learning

The goal of representation learning is to optimize machine learning algorithms to learn useful representation from the data for different predictors [8]. Neural networks are considered representation learning models since it usually encodes information into a discrete number of features. These features can be paired with, for example, linear layers to create a classifier. For example, as discussed in Section 1, a typical machine listening device is embedded with a feature extractor. This feature extractor will embed the low-level representation of the recorded signal (either time domain signal or time/frequency spectrogram) to a number of features. The output feature vector has lower dimension, summarizes audio information, and is capable of SEC. Thus, the feature extractor is car-

rying out representation learning to learn a good audio representation for SEC.

In this work, the feature extractor is expected to learn such representation that still maintains good SEC performance, while hiding the speech information to reduce the SD performance to preserve privacy.

2.3 Sound event classification & detection

Nowadays, people have more interest in having machines to assist in automated tasks. To do so, machine must be able to process information in the environment that it operates in. One of the sources of information is sound, and the capability of automatically comprehending and extracting audio information of machines is known as machine listening. One type of information that machine listening devices usually utilize is sound event. Sound event is the occurance of an incident that can be detected in the audio recording. Sound event classification (SEC) is the recognition of the active sound event or events in the audio recording [9]. In addition, sound event detection (SED) also detect the temporal activities of the sound events [9], [10].

A standard SEC/SED system usually consists of a feature extractor to calculate a latent representation of audio data. The output feature vector will be fed to another neural network model to carry out the classification/detection task. Even though the feature extractor is optimized with the classifier/detector to maximize the sound event information in the feature vector, theoretically other information can be embedded as well. This might also include speech and speaker's private information, which has been discussed in 1. Through some deep learning methods, these information can be recovered and misused, which raises a privacy concern for the users of the smart machine listening devices.

2.4 Voice activity detection

Voice activity detection (VAD) refers to the task where a model tries to detect whether speech signal is available and estimate the temporal activities of speech in the audio sample [11]. VAD is the first processing step for further speaker/speech processing tasks, such as speaker identification, speech recognition, etc., since it is more efficient to carry out lower level speaker/speech processing tasks on the segments where speech is available.

Therefore, decreasing the VAD capability of the output features means prevention of information leakage at a very early stage. Moreover, since VAD is relatively easy comparing to aforementioned lower level tasks, a privacy-preserving system with VAD is expected to work with more complex privacy tasks. Thus, a high level task like VAD is a good privacy scope for this research.

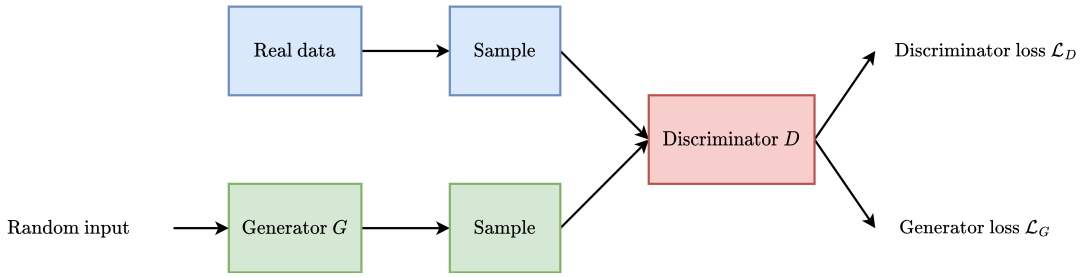

Figure 2.5. *GAN diagram. Figure adapted from Google Developers*

2.5 Adversarial learning

We are living in an era where machines possess human's language capability, creativity and imaginations. Machines can now generate human liked text and provide extensive answers and assistance to different questions, create images with any art style and any content that we want, and generate media that cannot be discriminated with real content. Many of these generative models originate from a simple system called GAN (generative adversarial network). Initially proposed by [12], GAN is a neural network system that creates an adversary setting between a generative model and discriminative model. The discriminative tries to predict whether a sample comes from real data distribution (original data) or model distribution (generated by the generative model). The generative model tries to generates counterfeit samples to fool the discriminator. The adversary continues throughout the training period until the discriminator can no longer adapt and discriminate between real and generated samples, and now the generator can generate samples which are very close to real data, i.e. the generated data belongs to the real data distribution. Figure 2.5 shows the basic structure of a GAN. Depending on the scenario where the discriminator is making predictions, discriminator loss and generator loss are defined. Denoting the discriminator loss as $\mathcal{L}_D$ and the generator loss as $\mathcal{L}_G$, we have the minimax criterion:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} \mathcal{L}_D + \mathbb{E}_{z \sim p_z(z)} \mathcal{L}_G \tag{2.2}$$

A very similar approach to GAN is proposed by [3] for domain adaptation. The motivation for [3] is to learn a feature extractor that is capable of a particular task regardless of either data distribution that a sample belongs to. Apart from the feature extractor, a domain classifier and a label predictor is presented in the system. Figure 2.6 illustrates the unsupervised domain adaptation system proposed by [3]. To train the domain adaptation feature extractor, a minimax objective during back-propagation is constructed using a component called gradient reversal layer (GRL). During backpropagation, GRL multiplies the gradient of the domain classifier loss to the feature extractor with a negative value λ:

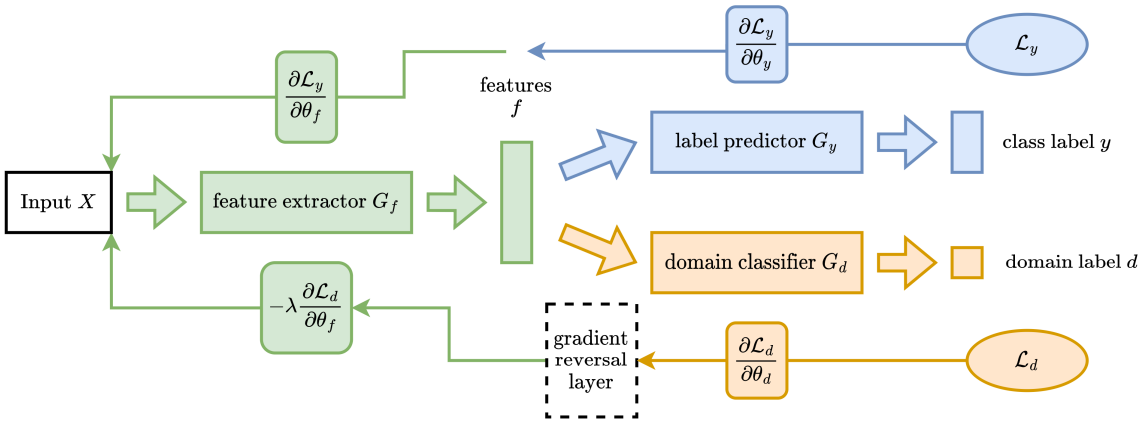

Figure 2.6. *The unsupervised domain adaptation system. Figure adapted from [3]*

$$\theta_f \longleftarrow \theta_f - \mu \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right) \tag{2.3}$$

$$\theta_y \longleftarrow \theta_y - \mu \frac{\partial \mathcal{L}_y}{\partial \theta_y} \tag{2.4}$$

$$\theta_d \longleftarrow \theta_d - \mu \frac{\partial \mathcal{L}_d}{\partial \theta_d} \tag{2.5}$$

where μ is the learning rate, θ_f, θ_y, θ_d are parameters of the feature extractor f, label predictor y, and domain classifier d, respectively. $\mathcal{L}_y$ is the loss of y and $\mathcal{L}_d$ is the loss of the d. This backpropagation process will ensure the optimization of y and d towards minimum, while maximizing $\mathcal{L}_d$ during training of f.

Reference [3] is an inspiration to many privacy-preserving audio systems for its modularity and wide applicability to various problem setups. The goal of privacy-preservation usually is to learn an audio representation that is capable of one processing task and able to hide sensitive speakers information. Thus, by setting the feature extractor and label classifier to the architectures required by the task and setting the domain classifier to an architecture for the defined privacy goal (speaker identification, speaker gender classifier, etc.), we can suppress the sensitive privacy information in the representation. The systems proposed in [1], [13], [14] and this work are based on the domain adaptation system by [3].

3. METHOD

3.1 System architecture

After data creation and preprocessing, we have the low level audio features x (log-mel spectrograms), sound event labels y, and speech labels s. The adversarial system is desired to correctly predict the sound event labels while hiding information about the speech labels.

The system setup consists of a feature extractor F, a sound event classifier C and speech discriminator D. Before D, a gradient reversal layer (GRL) is present to reverse the sign of the gradient to achieve the privacy goal. Figure 3.1 illustrates the setup of our adversarial system.

3.1.1 Feature extractor

The feature extractor F is a CNN-based neural network to embed the low-level audio feature x (log-mel spectrogram) to higher level latent audio representation z. For F, We adapt the *CNN6* architecture from [15] to match our input data and system setup. Figure 3.2 shows the architecture of the feature extractor

The architecture of F includes 4 convolutional blocks. Each block consists of a convolutional kernel of size 3×3, ReLU activation function and batch normalization. The number of convolutional kernels are 64, 128, 256 and 512, respectively. The first three convolutional layers are applied max pooling with kernel of size 2×2, and the last convolutional

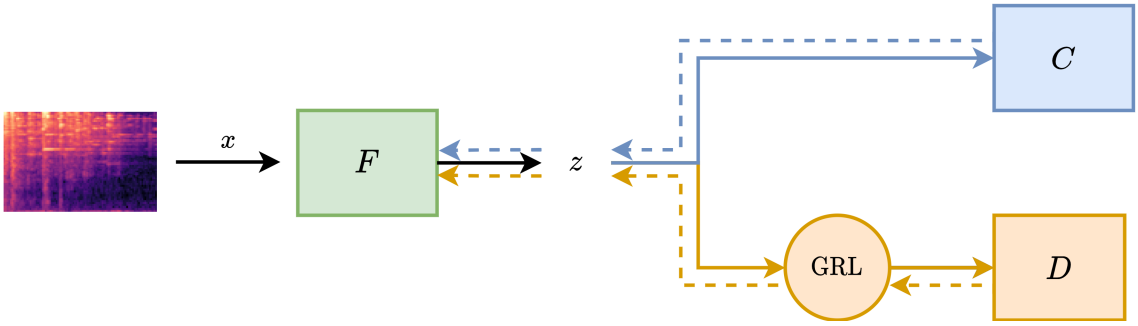

Figure 3.1. *Setup of the adversarial system*

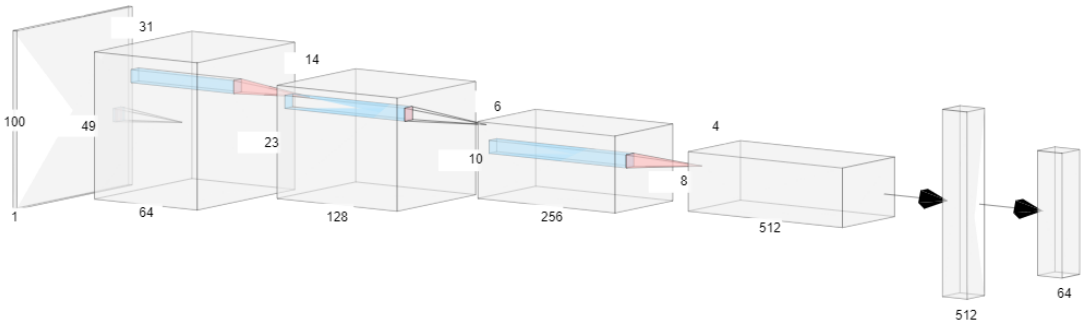

Figure 3.2. *Architecture of* F

layer is applied with global max pooling to embed 2D data into 1D vector. The 1D vector after max pooling is also fed through a fully connected layer to reduce the representation from 512 elements to 64 elements.

3.1.2 Sound event classifier

The task of the sound event classifier C is to predict the probabilities of each sound event that the sample contains, and determine the correct sound event based on the predicted results. C contains a fully connected layer with softmax activation function to map the feature vector of 64 elements to 3 elements, corresponding to the probability of each sound event.

3.1.3 Speech discriminator

Similar to C, the task of D is to output the probability whether speech exists in the sample. For the adversarial learning system to work properly, F and D should have comparable number of parameters. Therefore, D contains 4 fully connected layers, with output dimension of 48, 32, 16, and 1, respectively. The first 3 layers have LeakyReLU as the activation function, and the last layer have sigmoid as the activation function.

3.1.4 Training losses & Gradient reversal layer

The SEC loss $\mathcal{L}_C$ is defined as the cross entropy loss between the predicted probability of each sound event $\hat{y}_i = C(F(x))$ and the true value y_i:

$$\mathcal{L}_C = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathbb{X}} \sum_{i=1}^{N} y_i \log \hat{y}_i \tag{3.1}$$

The SD loss $\mathcal{L}_D$ is defined as the binary cross entropy loss between the prediction of speech presence $\hat{s} = D(F(x))$ and the real label s:

$$\mathcal{L}_D = -\mathbb{E}_{(\mathbf{x},\mathbf{s})\sim\mathbb{X}} \sum s \log \hat{s} + (1-s) \log(1-\hat{s}) \tag{3.2}$$

As proposed by [3], the gradient reversal layer multiplies the gradient of $\mathcal{L}_D$ to F with a negative value λ, hence updating the parameters of F with a minimax criterion (2.3-2.5). The λ value is not updated while back-propagation, but defined manually before the training process.

For the first 30 epochs, λ is set to 0 to create a supervised learning for F, C, and D. This is for all the components of the network to reach sufficient performance and enhance the effect of adversarial learning. After 30 epochs, λ increases with the following function, adapted from [3]:

$$\lambda_\beta = \frac{2}{1+\exp(-\gamma.\beta)} - 1 \tag{3.3}$$

where γ is fixed to 100, and β increases linearly from 0 to 1 after the first 30 epochs.

3.2 Training specifications

The training is conducted via PyTorch with CUDA support. The hardware used for training is NVIDIA Tesla V100 on TUNI's Narvi cluster.

Minibatch gradient descent with batch size of 64, momentum of 0.9 and learning rate of 0.01 is used. The training goes for maximum 5000 epochs. Early stopping condition is set on validation SD loss: If the loss does not increase after 50 epochs, the training is stopped and the best models are recorded.

4. RESULTS

4.1 Data

To be suitable for our work, the dataset should contain some mixtures of sound event and speech recordings. We artificially create the audio mixtures using samples from Freesound 50K dataset and LibriSpeech dataset.

4.1.1 Freesound 50K dataset

Freesound 50K dataset (FSD50K) [16] is a popular dataset for machine listening. FSD50K contains audio from multiple sound sources, such as human sounds, animals, objects, musical instruments, etc. In FSD50K, 51 197 audio clips from Freesound are available and manually labeled using 2000 classes drawn from AudioSet Ontology [17]. The data are provided in 2 splits: development and evaluation. The recordings are sampled at 44.1 KHz, and additional metadata about the description of the recordings and the possible labels are available. The dataset is licensed under CreativeCommons license. Figure 4.1 shows the example waveforms of 2 samples from FSD50K.

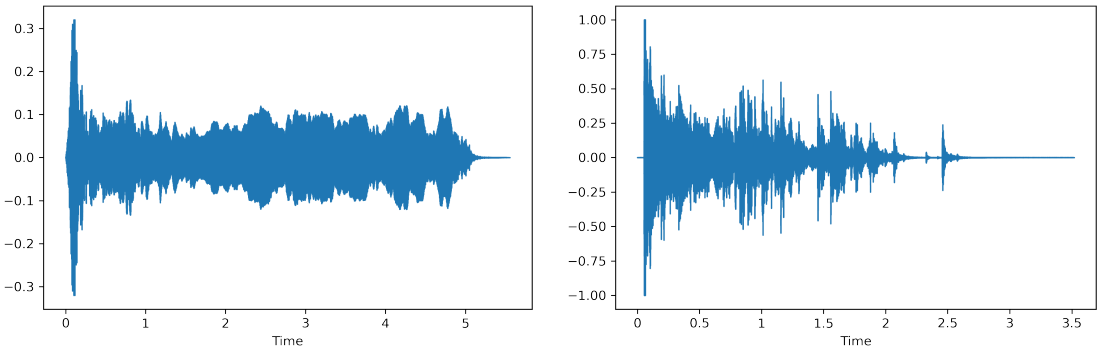

Figure 4.1. *Waveforms of 2 samples from FSD50K dataset. The sample on the left belongs to the Dog barking class, and the sample on the right belongs to the Glass breaking class.*

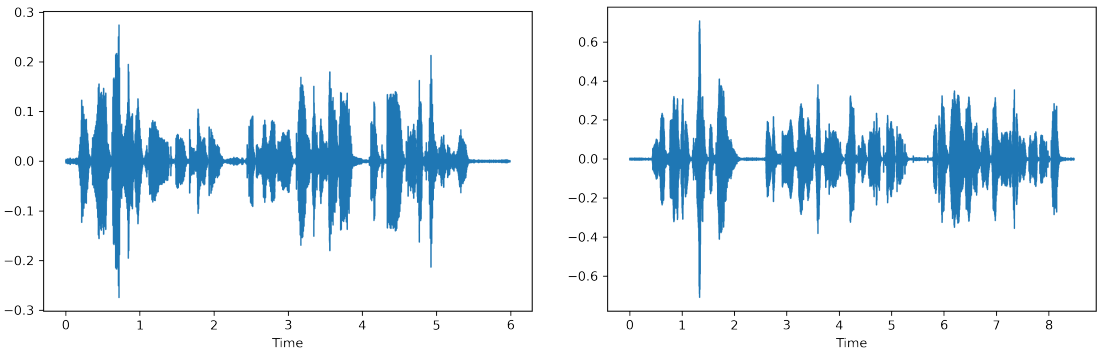

Figure 4.2. *Waveforms of 2 samples from LibriSpeech corpus. The sample on the left is recorded from a male speaker, and the sample on the right is recorded from a female speaker.*

4.1.2 LibriSpeech corpus

LibriSpeech corpus [18] includes around 1000 samples of English speech. The recordings are recorded from various male and female speakers reading English books. Multiple splits are provided according to the purpose and characteristics of the speech (whether the speech is "clean"). Additional data about the speaker and the book that the recording is extracted from is available. The recordings are sampled at 16 kHz, and distributed under CreativeCommons license. Figure 4.2 visualizes the waveforms of 2 example samples in LibriSpeech corpus.

4.1.3 Data creation

For the sound event data, we select the samples from the following 3 sound events: Dog barking, Glass breaking and Gun shot. The selection is due to the sufficient and similar numbers of samples between the 3 sound events. The samples from FSD50K's *development* split form our *dev* split, and the samples from FSD50K's *evaluation* split form our *test* split. To normalize the data, from all samples, we extract the most energetic one-second segment and subtract the elements by the mean and divide the elements by the standard deviation of the segment. After normalization, the segments are scaled with 0.05 to match the range $[0, 1]$.

We select the speech samples from LibriSpeech's *train-clean-100* split to merge with our *dev* split, and the speech samples from LibriSpeech's *dev-clean* split to merge with our *test* split. This is to maintain the distinction and the intended usage of each data split. The samples from LibriSpeech corpus are resampled from 16 kHz to 44.1 kHz to match the Freesound samples, then the similar normalization process is applied to the speech recordings.

To merge the sound event data and speech data, we add the two signals together in time

Table 4.1. *Number of unique speakers in dev and test splits and speech duration in seconds*

	dev	**test**
Male speakers	126	20
Female speakers	125	20
Speech duration (s)	588	176

Table 4.2. *Number of samples for each sound event in each split*

Sound event	**Training**	**Validation**	**Test**
Dog barking	374	40	122
Glass breaking	374	40	96
Gun shot	314	34	134

domain. Since adversarial learning tends to be less stable than usual supervised learning, we want to mitigate the potential problems that might affect the optimization process. Thus, the speech signals are attenuated by 5 db before merging to make the privacy-preservation task slightly simpler. In each sound event in each data split, half of the samples are chosen randomly to be added with speech. In the data splits in LibriSpeech corpus, the number of male and female speakers are balanced, so by choosing a similar number of samples from each speaker, we can also ensure the number of recordings from male and female speakers are balanced. 2 example samples from our dataset in the Glass breaking class is shown in Figure 4.3.

Since another data split is necessary for model selection, our *dev* dataset is divided into *train* and *val* with the ratio of $9 : 1$. The ratio constraint applies to the samples with speech and the samples without speech in each sound event; however, there are no further constraint on the gender of the speaker. Table 4.1 shows the number of unique speakers in the *dev* and *test* splits and speech duration, and Table 4.2 shows the number of samples of each sound event in *train*, *val*, and *test* splits.

4.1.4 Data preprocessing

Log-mel spectrogram is one of the most common low level representation of audio signal for machine listening systems. A spectrogram represents the data in both time and frequency domains; therefore, it conveys more information about different components of the audio signal. Representing the frequency bands in mel-scale is common since it is analogous to human's auditory system - humans do not perceive different frequencies bands linearly.

To calculate the log mel spectrograms of the features in our dataset as the low level in-

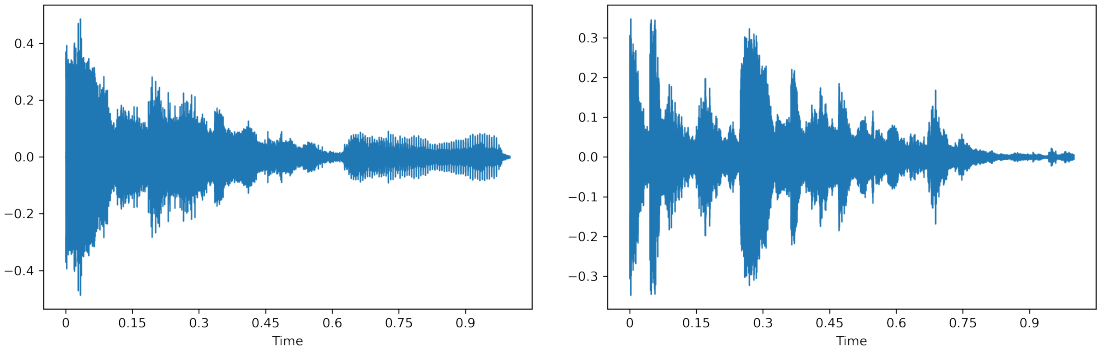

Figure 4.3. *Waveforms of the 2 samples from our dataset, in the Glass breaking class. The sample on the left are merged with speech. The sample on the right are not. Both are 1-second long and sampled at 44.1 kHz.*

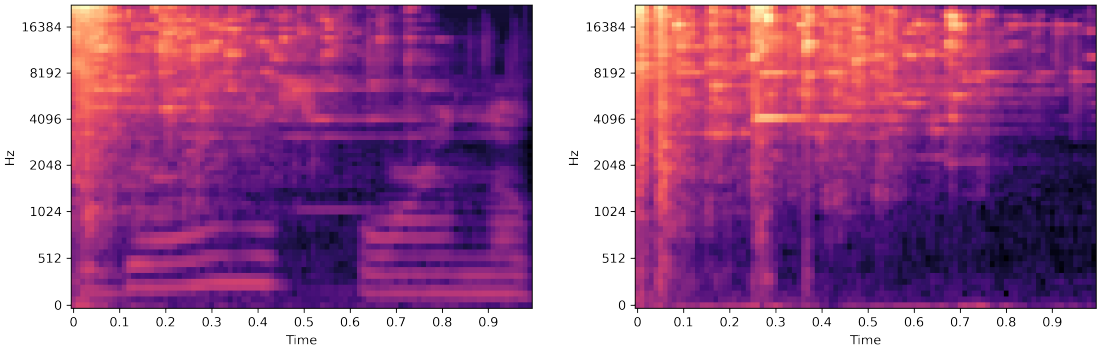

Figure 4.4. *The log-mel spectrogram of the presented samples from our dataset*

put feature for the network, we use short-time Fourier transform (STFT) with Hamming window of length 1411 and hop length of 441. The mel-scale is calculated using 64 mel filterbanks. We then calculate the mean and standard deviation of every bin in training data, and use such result for normalization of both *train*, *test*, and *val* data splits. Ultimately, log-mel spectrograms of size 64×100 are inputs of the system. Figure 4.4 illustrates the log-mel spectrograms of the previously presented samples from our dataset.

4.2 Evaluation criterion

The sound event classification performance is measured using classification accuracy. To verify privacy-preservation capabilities of F, we train a new speech discriminator D' using the output representation from F and report the accuracy and recall of D'. Each experiment is done 10 times, and the mean and standard deviation of the results are reported.

Table 4.3. *The results from the experiments of the baseline and adversarial learning setups. Each experiment is done 10 times and the mean and standard deviation of the results are reported in the format* $mean \pm std$.

Setup	$\mathbf{C}_{\text{accuracy}}$	$\mathbf{D}_{\text{accuracy}}$	$\mathbf{D}_{\text{recall}}$	$\mathbf{D'}_{\text{accuracy}}$	$\mathbf{D'}_{\text{recall}}$
Baseline	0.84 ± 0.01			0.79 ± 0.02	0.85 ± 0.02
Adversarial learning	0.84 ± 0.01	0.47 ± 0.03	0.52 ± 0.22	0.80 ± 0.02	0.82 ± 0.04

4.3 Baseline

The baseline is a sound event classification system that does not have any security measures. In the baseline system, F and C are optimized together in a supervised manner, and only $\mathcal{L}_C$ (Equation 3.1) is minimized.

4.4 Adversarial results

Figure 4.5 shows the training curves of the optimization process. The training SEC loss converges to a very small value, which indicates that adversarial learning process can reach good SEC performance. There is a lot of fluctuation in the validation SEC loss, but it is understandable with such low number of validation samples. The SD losses converge at a value of $0.69 \approx \log 2$, which is an ideal value in our setup:

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{(\mathbf{x},\mathbf{s})\sim\mathbb{X}} \sum s \log \hat{s} + (1-s)\log(1-\hat{s}) \\ &= -\frac{1}{N}((1\log 0.5 + (1-1)\log(1-0.5)) + (0\log 0.5 + (1-0)\log(1-0.5)) + ...) \\ &= -\frac{1}{N} N \log 0.5 \\ &= \log 2\end{aligned} \tag{4.1}$$

This ideal value is resulted from the fact that when D has no information about the presence of speech, its output will be 0.5 for all samples, which means total confusion. The loss curves indicate that the adversarial training process is successful, and throughout the training process, F has deceived D while still achieving good SEC capability.

Table 4.3 shows the results from the baseline adversarial setup. The accuracy of C in both the baseline and adversarial setups are very similar, which shows a little trade-off between utility and privacy-preservation of the feature extractor optimized by adversarial learning.

The accuracy and recall of D is very close to 50%, which is another evidence that during adversarial learning, F has successfully fooled D. On the other hand, after training D'

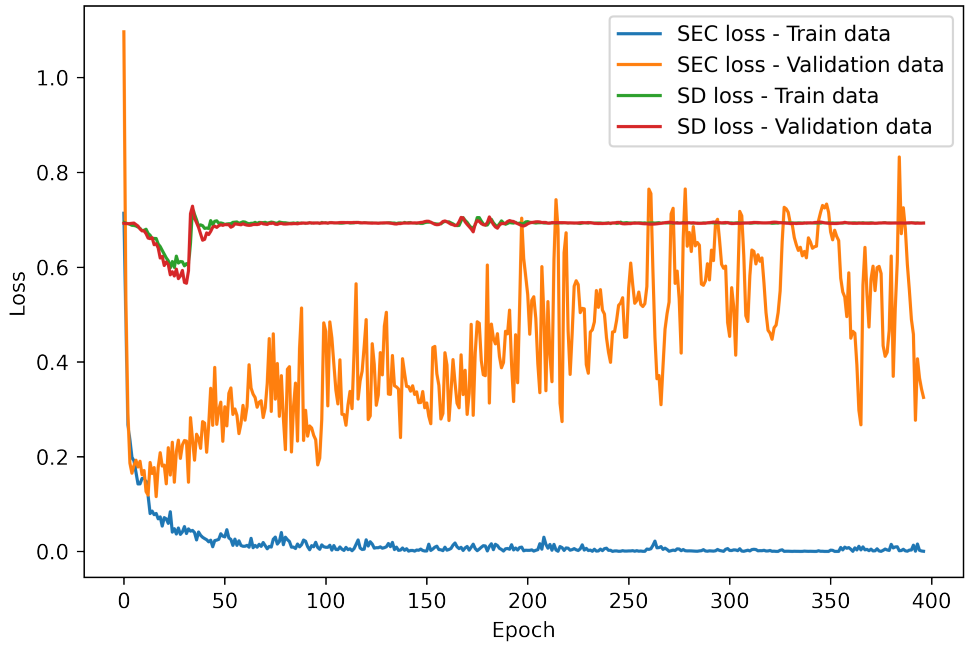

Figure 4.5. *The loss curves during the training process. The losses include the SEC loss on the training and validation data, and the SD loss on the training and validation data.*

on the output features, the performance of D' is close to the baseline. This indicates a similar problem to [14], where some residual privacy information can still be recovered and negatively affect the privacy-preservation capability of F.

Figure 4.6 shows a density curve for the distribution of the predictions of probabilities of speech by D', estimated by using a Gaussian kernel. A value close to the boundaries (either 0.0 or 1.0) indicates a highly confident prediction of D', while a value closer to the threshold value (0.5) shows that D' is less certain in its prediction. In both the baseline and adversarial learning system, there is not much overlap between the distributions of speech and non_speech samples. In addition, the predicted values concentrate around the boundaries, which means D' is quite certain in its prediction of speech, and a large amount of privacy information can still be recovered.

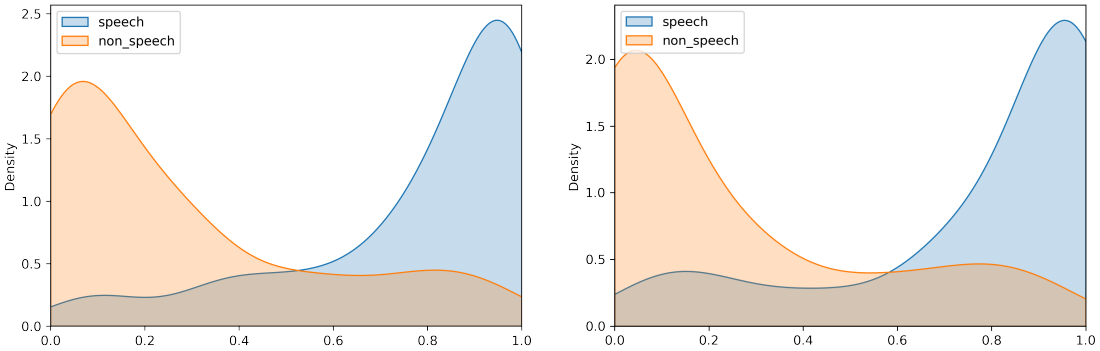

Figure 4.6. *The density curves of the predicted probabilities of speech from D' in the baseline and adversarial setups. The left is from the baseline, and the right is from the adversarial learning system.*

5. CONCLUSION

Thanks to the development of deep learning and neural networks, more and more automated devices for different tasks are introduced to everyday life. One of the most popular applications of these devices is machine listening, where devices can record audio and detect different sound events in the recording. Since human voice is also a common sound source, the information being transferred between the device and the cloud is likely to contain sensitive information can be extracted from speech signals and cause privacy threat to the users.

Adversarial network is currently a machine learning topic attracting a lot of research due to its extensive application in multiple tasks. Specifically, adversarial learning is one of the possible approaches to tackle the privacy problem faced by smart IoT devices. A system with domain adaptation approach [3] has the potential to conceal privacy information by minimizing the discriminability of the presence of such information.

In this thesis, we investigate the effectiveness of an adversarial learning into learning a privacy-preserving representation of audio signal for machine listening purposes. The representation should be capable of SEC, and at the same time contaning minimal information for SD. Inspired by [3], we create and adversarial setup between F, C, and D. The results show that utility-wise, no trade-off is found between the adversarial setup and the baseline setup. Privacy-wise, even though the adversarial learning process works as expected (D fails to discriminate between speech and non-speech samples), the residual privacy information can still be recovered, which leads to similar evaluation results in privacy preservation ability between the adversarial and baseline setup. Ultimately, basic adversarial learning is not yet able to achieve privacy preservation, and further modifications are necessary to achieve such goal.

REFERENCES

[1] J. M. Perero-Codosero, F. M. Espinoza-Cuadros, and L. A. Hernández-Gómez, "X-vector anonymization using autoencoders and adversarial training for preserving speech privacy," *Computer Speech & Language*, vol. 74, p. 101 351, 2022, ISSN: 0885-2308.

[2] J. Williams, J. Yamagishi, Paul-Gauthier, C. Valentini-Botinhao, and J.-F. Bonastre, "Revisiting Speech Content Privacy," in *Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication*, 2021, pp. 42–46.

[3] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 1180–1189.

[4] K. Zainal-Mokhtar and J. Mohamad-Saleh, "An oil fraction neural sensor developed using electrical capacitance tomography sensor data," *Sensors*, vol. 13, no. 9, pp. 11 385–11 406, 2013, ISSN: 1424-8220.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, 2019, ISSN: 2076-3417.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, ISSN: 0162-8828. DOI: 10.1109/tpami.2013.50. [Online]. Available: https://doi.org/10.1109/TPAMI.2013.50.

[9] S. Adavanne, H. Fayek, and V. Tourbabin, "Sound event classification and detection with weakly labeled data," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, Oct. 2019, pp. 15–19.

[10] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[11] T. Bäckström, O. Räsänen, A. Zewoudie, *et al.*, *Introduction to Speech Processing*, 2nd ed. 2022.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014.

[13] Z. Meng, J. Li, Z. Chen, *et al.*, “Speaker-invariant training via adversarial learning,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5969–5973.

[14] B. M. L. Srivastava, A. Bellet, M. Tommasi, and E. Vincent, “Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion?” In *Proc. Interspeech 2019*, 2019, pp. 3700–3704.

[15] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[16] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, “Fsd50k: An open dataset of human-labeled sound events,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, pp. 829–852, Feb. 2022, ISSN: 2329-9290.

[17] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, *et al.*, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.