Speaker Diarization Using Improved SincNet Models to Extract Speaker Embeddings

by

Mohammad Dib

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Speaker diarization is the process of identifying who spoke when in an audio stream, and it is applied in many fields, such as information retrieval and psychotherapy. Speaker embeddings extraction is a crucial step in any diarization system, where the goal is to extract highly discriminative speaker embeddings (d-vectors). Most of the existing methods are based on deep neural networks (DNNs) and they rely on engineered features, which may not guarantee optimal performance for all cases. This led to the development of the SincNet model, which can effectively and efficiently process raw input audio signals. The SincNet model was successfully used to perform embeddings extraction in a speaker diarization system, where it resulted in a high diarization performance. Its successor, the AM-SincNet model, which combines SincNet with an improved loss function, outperformed the standard SincNet on the speaker diarization task. This shows the importance of enhancing the loss function of SincNet to achieve better diarization performance.

Thus, the goal of this thesis is to improve the ability of the SincNet model to extract discriminative embeddings such that it results in a better diarization performance by experimenting with different architectures and state-of-the-art loss functions. In this thesis, 16 different SincNet based models were proposed as follows: four models that combine the SincNet architecture with four different loss functions, six models that combine the Res-SincNet architecture (a recently proposed architecture) with six different loss functions, and six models that combine the Res-SincNet-FC architecture (proposed in this thesis) with six different loss functions.

The results show that the proposed MV-AM-SincNet model gives the best diarization performance compared to all the models discussed in this thesis. This shows the high capability of the MV-Softmax loss at extracting highly discriminative embeddings compared to the other losses. Additionally, the speaker recognition performance was reported, since all the models were trained for speaker recognition before being applied in speaker diarization. It was found that the proposed Res-SincNet-FC architecture resulted in the lowest frame error rate (FER) when combined with the different loss functions, where the D-Res-SincNet-FC and Arc-Res-SincNet-FC achieved the lowest FER. The Visualization of the extracted embeddings and the diarization output of the MV-AM-SincNet model showed its ability to extract highly discriminative embeddings. However, the visualization showed that having a large number of overlapping segments and/or small speaker segments impacts the diarization performance negatively.

In this thesis, significant improvements on the SincNet model were made, which assists in achieving higher speaker recognition and diarization performance, where the raw audio signals were processed efficiently and effectively, without the need for feature engineering.

Acknowledgements

I would like to greatly thank my supervisor, Professor Otman Basir, for his support, guidance, and encouragement throughout my MASc, and for giving me the opportunity to work under his supervision.

Also, I would like to thank my thesis committee members, Prof. Zhou Wang and Prof. Mark Crowley, for their valuable time in reviewing my thesis.

I would like to express my sincere gratitude to my family for their unconditional love and support. Finally, I would like to thank my friends and all the people who helped me throughout my studies.

Dedication

This is dedicated to my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech is the primary method of communication among humans. Thus, it would be highly beneficial to develop technologies that are capable of processing and analyzing human speech. One such important technology is automatic speech recognition (ASR) which takes speech signal as an input and then converts it into text [3]. This speech-to-text conversion is called transcription and it can be used in a wide range of applications. For example, in a medical setup, practitioners can save valuable time with patients using automatic transcription as a substitute for manual note-taking [4]. Additionally, transcription can be found in finance firms, government institutions, and in the automotive industry [4]. Multiple large companies are involved in advancing speech recognition technology and distributing it through their products. Major examples include Alexa from Amazon and Siri from Apple, where both products serve as voice-enabled personal assistants. Moreover, Google provides voice-enabled commands for mobile phones and internet search. The speech recognition technology has a huge market size. According to a 2021 report by IBISWorld, which is a company that provides in-depth research on industries worldwide, the revenue of the speech and voice recognition industry in the US had an estimated annual growth of 16.3% over the past five years to reach $4.2 billion in 2021 [4]. Furthermore, according to Grand View Research, Inc., a company that provides market research and consulting services, the revenue of the global speech and voice recognition industry is expected to reach $31.82 billion by 2025 with a compound annual growth rate of 17.2% [5]. Thus, developing highly accurate speech recognition systems is crucial due to their useful applications and economical value.

While research aiming to improve ASR systems has exploded due to its wide range of applications, some challenges facing the speech recognition community still persist. For example, the performance of speech recognition systems can be negatively impacted in the presence of multiple speakers, since the speech from the different speakers may overlap, causing a confusion with the ASR system. Consequently, this confusion results in erroneous transcription. This problem can be solved by a process called speaker diarization, which aims to identify who spoke when in an audio stream. More precisely, speaker diarization is the process of segmenting an input audio stream according to speaker identities, such that each segment corresponds to a unique speaker. Therefore, diarization can be used as a front-end processing step to improve the performance of ASR systems in scenarios with more than one speaker. Besides improving the performance of ASR systems, speaker diarization has other applications, including information retrieval from audio data involving multiple speakers (e.g. conferences). Additionally, diarization has healthcare-related applications, such as psychotherapy, where it automatically acquires speaker annotations from therapy sessions between therapists and patients, which can help the therapists perform further analysis [6]. Thus, it is important to develop highly accurate and efficient speaker diarization systems.

Speaker embeddings (d-vectors) extraction is a crucial step in speaker diarization, where the goal is to extract speaker-related features that facilitate the distinguishing of different speakers [7]. Currently, deep neural networks (DNNs) achieve better embeddings extraction compared to older methods, such as i-vectors [8]. Most of these DNN models rely on engineered features, such as MFCC, to extract speaker embeddings. However, relying on engineered features may not be appropriate for all applications, and may prevent the network from achieving optimal performance [1]. Thus, SincNet [1] was proposed to efficiently and effectively process raw input signals, where it outperformed the usage of engineered features. Then, AM-SincNet [9] and AF-SincNet [10] improved the performance of SincNet by utilizing more discriminative loss functions. On the other hand, Res-SincNet [2] enhanced the performance of SincNet by modifying its architecture, where it was combined with residual layers.

In [8] and [11], the authors used the SincNet model [1] and the AM-SincNet model [9] as speaker embeddings extractors in speaker diarization systems, respectively. The AM-SincNet model, which replaces the softmax loss in the original SincNet model with a more discriminative loss called AM-Softmax, achieved a better diarization performance compared to the original SincNet model. This indicates the impact of using a more discriminative loss function on improving the diarization performance.

Therefore, this thesis aims to improve the ability of SincNet based architectures to extract highly discriminate embeddings by incorporating different state-of-the-art loss functions, which is crucial to achieving an improved diarization performance.

1.1 Motivation

Speaker diarization has several important applications, which require high diarization accuracy. Discriminative speaker embeddings extraction plays a crucial role in achieving high diarization performance. Although there are many speaker embeddings extraction methods used for speaker diarization, they face some issues that need to be addressed. Most of the existing speaker embeddings extractors in speaker diarization employ engineered features, such as MFCC, instead of using the raw input waveform, which do not always guarantee optimal performance. Additionally, the existing speaker embeddings extractors which can process raw input audio signals efficiently and effectively, such as the SincNet [1] and the AM-SincNet [9] models, can be further improved by utilizing more discriminative loss functions. Moreover, the literature lacks the application of an improved SincNet architecture, such as Res-SincNet [2], as speaker embeddings extractor in a speaker diarization system. Thus, there is a significant room to further enhance the SincNet model [1] to perform better speaker embeddings extraction in order to achieve an improved diarization performance. This can be achieved by combining the SincNet architecture with a more discriminative state-of-the-art loss function. Another approach worth investigating would be to first improve on the existing SincNet architecture, and then combine it with different loss functions.

1.2 Problem Statement

The main goal of this thesis is to develop a speaker embeddings extractor to be subsequently used in a speaker diarization system, where the embeddings extractor can process raw input waveforms efficiently and effectively through the use of the SincNet architecture. The ability of the SincNet model to extract highly discriminative speaker embeddings will be enhanced by incorporating different state-of-the-art loss functions and combining them with different SincNet-based architectures.

1.3 Objectives

The process of improving the performance of the SincNet model by utilizing different state-of-the-art loss functions and different SincNet-based architectures can be formulated into the following objectives:

1. Conduct a literature review on the current state-of-the-art loss functions and identify multiple potential candidates.

2. Combine the chosen state-of-the-art loss functions with the SincNet architecture [1] and compare their speaker recognition and diarization performance.

3. Combine the chosen state-of-the-art loss functions with the Res-SincNet architecture [2] and compare their speaker recognition and diarization performance.

4. Improve the Res-SincNet architecture such that it becomes more suitable to be combined with the chosen state-of-the-art loss functions.

5. Combine the chosen state-of-the-art loss functions with the improved Res-SincNet architecture and compare their speaker recognition and diarization performance.

6. Analyze the performance of the SincNet-based model that gave the best diarization performance by visualizing its extracted speaker embeddings.

1.4 Contributions

This thesis contributes to the speaker embeddings extraction step in speaker diarization systems, where speaker embeddings are extracted from the raw input waveform directly without the need for a features engineering step. This is achieved by improving on the SincNet model, which can process raw input signals efficiently and effectively, by utilizing different state-of-the-art loss functions and different architectures. Additionally, the speaker recognition performance of the SincNet model was considered and improved on in this thesis, which is important for speaker diarization, since the speaker embeddings extractors are first trained to perform speaker recognition.

Firstly, the SincNet architecture [1] was combined with four distinct state-of-the-art loss functions. Most of the proposed models gave a similar diarization and recognition performance to the existing SincNet models, but the proposed MV-AM-SincNet model gave superior diarization performance compared to all of the other SincNet models. Moreover, experiments on the input shift size showed that it can be increased from 10 ms, which is the value used in the original SincNet paper [1] and other related papers, to 50 ms such that the processing time is reduced without negatively impacting the performance.

In addition, the Res-SincNet architecture [2] was combined with six different state-of-the-art loss functions. However, the Res-SincNet architecture was not suitable to be combined with the different losses due to the usage of the global average pooling layer. Thus, the Res-SincNet-FC architecture was proposed, which is more suitable to be combined with the different losses, since the global average pooling layer was replaced with a fully connected layer. Finally, the Res-SincNet-FC architecture was combined with six distinct state-of-the-art loss functions. The proposed models based on the Res-SincNet-FC architecture gave a significantly improved speaker recognition performance compared to the models based on the SincNet architecture [1], where the proposed Arc-Res-SincNet-FC and D-Res-SincNet-FC outperformed all of the other SincNet models on the speaker recognition task.

1.5 Thesis Outline

Chapter 2 gives a literature review on the existing SincNet based models, loss functions in general, and the chosen state-of-the-art loss functions that will be combined with the different SincNet architectures. Chapter 3 presents the proposed models, where different SincNet based architectures were combined with different state-of-the-art loss functions. Chapter 4 goes through the experimental setup and the results achieved in this thesis. Chapter 5 concludes the thesis and suggests future ideas to extend the work presented in this thesis.

Chapter 2

Background and Literature Review

In this chapter, I will go through the literature review of this thesis. Firstly, I will talk about the speaker diarization and recognition. Secondly, I will go over the existing SincNet models, where I will review: the SincNet model, the existing methods that improved the speaker recognition performance of SincNet, and the current usage of SincNet to perform speaker diarization. After that, I will go through an overview of loss functions. Finally, I will thoroughly discuss the chosen state-of-the-art loss functions that will be combined with the SincNet model.

2.1 Speaker Diarization and Recognition

The main goal of this thesis is to develop a highly discriminative speaker embeddings extractor for a speaker diarization system. However, the speaker embeddings extractor must be first trained to perform speaker recognition, where the extractor will learn how to extract discriminative features from input data. Thus, this thesis focuses on two tasks, speaker diarization and speaker recognition, which will be discussed in this section.

2.1.1 Speaker Diarization

Speaker Diarization addresses the question "who spoke when" in an audio file. Speaker Diarization has several important applications, such as the analysis of conference meetings and phone calls. Generally, speaker diarization systems are comprised of four stages [12]. The first stage is the speech segmentation, which removes the non-speech parts in the input audio signal, and then segments the speech parts into short segments where it is assumed that each segment has one speaker [12]. The second stage is the speaker embeddings extraction, where audio-relevant features, are extracted from the segments [12]. The third stage is clustering, where the extracted embeddings from the previous stage are clustered for each unique speaker [12]. Finally, a re-segmentation stage, which is optional, is used sometimes to refine the results from the clustering stage in order to improve the performance of the system [12].

The most widely used embedding methods are i-vectors, d-vectors, and x-vectors. I-vectors are based on Gaussian Mixture Models (GMM), and joint factor analysis [7]. On the other hand, d-vectors and x-vectors are based on deep neural networks (DNNs), where both methods outperform the i-vectors method, which shows the strength of deep neural networks for speaker embedding extraction [7]. Speaker embeddings extraction using DNNs is performed by first training the network to perform speaker recognition in a supervised manner [7]. Then, the pre-trained network can be used to extract d-vectors, where the d-vectors will be the output of the second last layer or the bottleneck layer in the pre-trained model [7]. The goal is to achieve highly separable and discriminative d-vectors between different speakers, which is achieved by employing discriminative loss functions [7].

2.1.2 Speaker Recognition

Speaker recognition has multiple important applications such as voice authentication and security [1]. Speaker recognition aims to recognize a speaker from a given set of speakers based on an input speech signal. Before the huge advancements in DNNs, the i-vector method, which derives features from audio signals and classifies them using different approaches, such as probabilistic linear discriminant analysis (PLDA) [13], used to be the state-of-the-art approach for performing speaker recognition [10]. Most of the current state-of-the-art DNN based speaker recognition methods employ hand-crafted features, such as FBANK, in their implementation [1]. However, these hand-crafted features may result in a sub-optimal performance due to inherited characteristics from the way they are calculated [1]. Thus, it would be beneficial to use raw speech signals instead of engineered features as the input to the DNN models. A good candidate to process raw speech signals is the SincNet model [1], which can perform this task efficiently and effectively.

2.2 Deep Neural Networks for Speaker Recognition and Diarization

Deep Neural Networks (DNNs) achieve state-of-the-art performance on both speaker recognition and diarization. Most of the DNNs rely on hand-engineered features, such as MFCC coefficients, to perform speaker recognition [1]. However, using handcrafted features may not be suitable for every speech-related problem, and they can negatively affect the learning of important narrow-band speaker features [1]. To overcome these issues, some authors proposed directly using raw signals as the input to the network, where Convolutional Neural Networks (CNNs) are the most commonly used networks to perform such tasks [1]. CNNs are widely used to process raw input waveform due to their useful properties, such as weight sharing and local filtering, which facilitate the learning of robust and invariant features [1]. Even though CNNs are suitable for processing raw input signals, their input layer is problematic, since it handles high dimensional data, and it is susceptible to vanishing gradient issues, particularly in the case of deep networks [1]. Moreover, the learned filters are noisy, uninterpretable, and do not seem to represent the speech waveform efficiently [1].

Thus, the SincNet architecture was introduced to efficiently and effectively process raw input waveform by using a sinc layer as the input layer to the CNN [1]. The sinc layer will convolve the input with sinc filters, which corresponds to a band-pass filter in the frequency domain, where the network is trained to learn the lower and upper cutoff frequencies of the sinc filters [1]. This allows the SincNet model to achieve high speaker recognition performance, while also facilitating fast convergence.

In this section, I will focus on the SincNet model. Firstly, I will give an overview of the SincNet model. Secondly, I will discuss how different authors worked on improving the performance of SincNet on speaker recognition. Finally, I will talk about the usage of SincNet in speaker diarization.

2.2.1 SincNet for Speaker Recognition

As mentioned previously, the SincNet architecture [1] is simply a Convolutional Neural Network (CNN), but with an input layer that consists of sinc filters. The sinc layer will perform a set of convolution operations in the time domain between the input raw speech signal and a set of sinc filters, which can be represented by the following equation:

$$y[n] = x[n] * g[n, f_1, f_2] \tag{2.1}$$

where $x[n]$ is the input raw speech signal, $g[n, f_1, f_2]$ is a sinc filter in the time domain, which corresponds to a rectangular band-pass filter in the frequency domain, where f_1 and f_2 are learnable parameters, and they represent the lower and upper cutoff frequencies, respectively, and $y[n]$ is the filtered output of the sinc layer. It is worth noting that if a standard convolutional layer is used as the input, then the network will learn every single element of the input filters; however, if the sinc layer is used, then the network will only learn the lower and upper cutoff frequencies of the sinc filters, which results in a significant reduction in the learnable parameters of the network [1]. The band-pass filter $g[n, f_1, f_2]$ in the frequency domain can be designed by subtracting two low-pass filters, as follows:

$$G[f, f_1, f_2] = rect(\frac{f}{2f_2}) - rect(\frac{f}{2f_1}) \tag{2.2}$$

where $rect(.)$ is the rectangular function in frequency domain. Then, applying inverse Fourier transform [14] on $G[f, f_1, f_2]$ results in the time domain representation, which is given by:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n) \tag{2.3}$$

where $sinc(x) = sin(x)/x$ is the sinc function.

The authors of SincNet [1] chose to initialize the cutoff frequencies of the input sinc filters with mel-scale filter-bank, instead of randomly initializing them, since this facilitates the allocation of larger number of filters in the lower region of the spectrum, which contains a lot of important indications regarding the speakers identities. The sinc layer only learns the bandwidth of the band-pass filters, while further layers in the network learn the filter gains by assigning different significance to the output of the sinc filters [1].

Additionally, in [1], windowing was applied on the sinc filters to reduce their band-pass ripples and increase their stop-band attenuation. The SincNet authors used Hamming window [15], since it encourages strong frequency selectivity. The Hamming window is defined as:

$$w[n] = 0.54 - 0.46 \cdot \cos(\frac{2\pi n}{L}) \tag{2.4}$$

where L is the filter length. The since filters after the application of the Hamming window is defined as:

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n] \tag{2.5}$$

The sinc filters g do not result in phase distortion, since they are symmetric [1]. Additionally, the symmetry of the filters can play a huge role in reducing the required computations, because the results can be calculated using a single half of the filter. Then, the calculated half can be used to derive the other half [1].

All the computations related to SincNet are differentiable, and all the SincNet parameters, including the lower and upper frequencies of the sinc filters, can be collectively optimized using gradient-based optimization methods, such as Stochastic Gradient Decent (SGD) [1].

The SincNet model [1] has three major properties:

1. SincNet has fast convergence, since it concentrates only on learning filter parameters that hugely affect the performance of the model, which simplifies the filters learning process
2. SincNet has significantly fewer parameters in its input layer compared to standard CNN, since the input layer of SincNet only learns the lower and upper cutoff frequencies, instead of learning every element of the filter as in standard CNNs. This means that the number of parameters in the sinc layer will not depend on the length of the filter, which facilitates the learning of highly selective filters without increasing the number of learnable parameters
3. SincNet learns highly interpretable filters in its input layer compared to other methods, which allows human inspection in order to understand what the network is learning

The authors of the SincNet model [1] designed their model to perform speaker recognition. They used two datasets with a different number of speakers in their experiments. It is worth noting that the authors of [1] used a realistically challenging setup for their experiments, where they used a relatively small number of training data per speaker, and they used test sentences with relatively short duration.

The authors of [1], compared the SincNet model with multiple other systems. They compared the SincNet with a standard CNN that takes raw signal as its input. Moreover, they compared SincNet with other neural network models that use commonly used hand-crafted features, in specific, they compared SincNet with both MFCC and FBANK features. Also, the authors compared the different models on the speaker verification task, where they extracted the d-vectors using the trained model, and they involved the i-vector method in their comparison.

The results showed the SincNet model [1] was able to effectively learn highly selective filters that can detect narrow-band speaker features, such as pitch and formants, which are important to successfully perform speaker recognition. Furthermore, the SincNet model [1] gave a superior speaker recognition performance compared to using standard CNN or hand-crafted features. The SincNet model [1] also outperformed the other models including i-vector on the speaker verification task, which indicates its strong ability to learn better speaker embeddings (d-vectors) compared to the other models.

Other SincNet Applications

The authors of [1], originally proposed the SincNet model to perform speaker recognition. However, in [16], they used the SincNet model to perform speech recognition, where they considered two datasets (one with close-speech and the other with distant-speech), and different conditions (i.e. clean and noisy). The results in [16] showed that the SincNet model gave a superior performance on the speech recognition task compared to the other models (a CNN with raw input and another CNN with FBANK features), while also having a faster convergence and more interpretable filters.

Other audio-related applications of the SincNet model include the following. In [17], SincNet was used in an emotion recognition system, where it outperformed the other models that utilize engineered features. Moreover, in [18], SincNet was successfully used to perform speaker counting in crowded conditions, where it gave better performance compared to methods that use hand-crafted features. Additionally, in [19], SincNet was effectively applied to perform acoustic model adaptation from speech of adults to speech of children under automatic speech recognition setting. Furthermore, in [20], SincNet was used in a multi-task learning system, where the goal was to jointly estimate the age and cognitive decline of a person through speech signals.

2.2.2 Improved SincNet Models for Speaker Recognition

In this section, I will discuss how different authors improved on the speaker recognition performance of SincNet, where two methods (AM-SincNet and AF-SincNet) modified the loss function of the SincNet model, while one method (Res-SincNet) modified the architecture of the SincNet model.

AM-SincNet

The authors of [9] improved on the SincNet model [1] by replacing the standard softmax loss with Additive margin softmax (AM-Softmax) loss [21], and they called their proposed model AM-SincNet. The AM-Softmax loss [21] gives better performance compared to the standard softmax loss, since it facilitates the learning of more discriminative features by imposing an additive margin to increase the angular distance between the different classes, which reduces intra-class variance and increases inter-class variance. The AM-Softmax loss [21] is defined as:

$$L_{AM} = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{s\cdot(\cos\theta_{y_i} - m)}}{e^{s\cdot(\cos\theta_{y_i} - m)} + \sum_{j=1, j\neq y_i}^{c} e^{s\cdot\cos\theta_j}} \tag{2.6}$$

where s is a scale parameter, m is the margin parameter that is tuned to achieve the desired performance, n is the number of samples in a given batch, c is the number of classes, j corresponds to the j-th class, and $\cos\theta_{y_i}$ is defined as:

$$\cos\theta_{y_i} = \frac{\langle \vec{W}_{y_i}^{T}, \vec{f}_i \rangle}{\|\vec{W}_{y_i}\| \|\vec{f}_i\|} \tag{2.7}$$

where f_i is the input to the classification layer corresponding to the i-th samples, W_{y_i} represents the weights connected to the ground-truth class y_i.

The authors of AM-SincNet [9] considered the same experimental setting used in the original SincNet paper [1], and they compared the AM-SincNet and SincNet models performance on the speaker recognition task, where they compared the results based on the Frame Error Rate (FER). Also, they fixed the s parameter in the AM-Softmax loss to 30, while they run different experiments using different m parameter values in the range [0.35, 0.8] with 0.05 increments, to understand the effect of the m parameter on the model performance. The AM-SincNet model gave a significantly better performance compared to the SincNet model, which shows the importance of using a discriminative loss function to achieve an enhanced performance [9]. Also, they found that most m values gave around the same improved performance, but for all m values the AM-SincNet outperformed the SincNet model.

AF-SincNet

Another modification on the SincNet loss function was proposed by [10], where they replaced the softmax loss with the additive angular margin (ArcFace) loss [22], and they called their model AF-SincNet. The ArcFace loss [22] is capable of effectively learning highly discriminative features due to the imposed angular margin, which encourages stronger intra-class compactness and inter-class separability. It is worth noting that the ArcFace loss is numerically similar to the AM-Softmax loss, but the ArcFace loss has a better geometrical interpretation and can result in a better performance [22]. The ArcFace loss [22] is defined as:

$$L_{Arc} = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{s\cdot\cos(\theta_{y_i}+m)}}{e^{s\cdot\cos(\theta_{y_i}+m)}+\sum_{j=1,j\neq y_i}^{c}e^{s\cdot\cos\theta_j}} \tag{2.8}$$

where the variable names have the same meaning as in Eq. 2.6.It can be seen that the AM-Softmax loss (defined in Eq. 2.6) and the ArcFace loss (defined in Eq. 2.8) have a very similar formulation, but the margin parameter m is directly added to the angle θ_{y_i} for the case of ArcFace, as opposite to AM-Softmax, which adds the margin parameter m to the cosine of the angle θ_{y_i}.

The authors of AF-SincNet [10] performed their experiments under the same settings of the SincNet paper [1], where they compared the performance of different models on the speaker recognition task. In [10], the authors based their comparisons on two performance measures, which are the Frame Error Rate (FER) and the Sentence Error Rate (SER). For the AF-SincNet model, the authors fixed the s parameter of the ArcFace loss to 30, while they performed hyper-parameter tuning for the m parameter, where they found that $m = 0.5$ results in the best performance. In [10], it was found that the AF-SincNet gives the best speaker recognition performance compared to the other models.

Res-SincNet

Both of AM-SincNet [9] (discussed in Section 2.2.2) and AF-SincNet [10] (discussed in Section 2.2.2) improved the performance of the SincNet model by modifying its loss function. However, the authors of Res-SincNet [2] improved the performance of the SincNet model by modifying its architecture. In [2], the authors replaced the fully connected layers that come after the convolutional layers in the SincNet model with residual layers [23], where this will improve model stability and performance, while also reducing overfitting and the number of parameters.

The experimental setup in [2], used the same datasets and the same evaluation metrics as in [1], but they improved the training optimization in their experiments, where they used early stopping instead of a fixed number of epochs, and a learning rate scheduler to reduce the training rate instead of a fixed learning rate. Additionally, the authors of Res-SincNet [2], tried various combinations of network hyperparameters, such as the number of filters in the sinc and convolutional layers, using an automatic hyperparameter optimization method called tree Parzen estimators [24], in order to study the effect of different parameters on the model performance. They found that some parameters had a greater influence on the Res-SincNet performance, for example, the stride and window size of the sinc layer were more important than that of the convolutional layer. Thus, the authors of [2] performed a thorough analysis of the window size and stride of the sinc layer by trying different combinations. They found that using a smaller sinc filter stride results in better performance, while the sinc window size had less impact on the Res-SincNet performance. Therefore, the authors suggested reducing the sinc window size which will decrease the computational requirements without heavily affecting the performance. The experiment in [2] showed that the Res-SincNet model gives a better FER performance compared to the SincNet model.

2.2.3 The Application of SincNet in Speaker Diarization

The authors of [8] used the SincNet model to perform speaker embeddings (d-vectors) extraction in a diarization system. They first trained the model to perform speaker recognition using an out-domain dataset, then they used the trained model as a d-vectors extractor to perform speaker diarization on an in-domain dataset. The authors of [8] tried extracting the speaker embeddings from different stages in the SincNet model, where they found that extracting the embeddings from the last hidden layer results in the best performance. Also, they found that using average pooling to get the segment level embeddings is better than max pooling. They tried reducing the dimensions of the extracted embeddings using principal component analysis (PCA), which improved the diarization performance. In [8], the authors focused on speaker embeddings extraction, hence they used the ground-truth speaker segments in their experiments. The authors of [8] employed length normalization before the clustering stage, where they used spherical K-means clustering algorithm [25] in their implementation. In [8], the SincNet model was compared to the i-vector method, in which it was found that extracting speaker embeddings using the SincNet model results in a better diarization performance.

The same authors of [8] extended their work in [11], where they used AM-SincNet instead of SincNet to extract speaker embeddings. The results in [11] showed that the AM-SincNet model achieved the best diarization performance compared to SincNet and i-vector method. The superior diarization performance achieved by the AM-SincNet can be attributed to its utilization of the AM-Softmax loss, which facilities the learning of highly discriminative features (embeddings), and hence resulting in a better diarization performance.

2.3 An Overview of Loss Functions

Loss functions (also called objective functions) are used to calculate the error between the predicted and the actual output [7]. Therefore, a neural network will work on minimizing this error, which is done using the optimizer [7]. To achieve high recognition performance, it is important to ensure that the learned features are both separable and discriminative, where this can be achieved through the use of discriminative loss functions [7]. Examples of discriminative loss functions are the contrastive loss and the triplet loss which gave a high performance in both face recognition and speaker verification tasks [7]. Both of these loss functions map the input into a feature space, where they minimize the intra-class variance and maximize the inter-class variance [7]. It is apparent that a good loss function should be capable of minimizing the intra-class variance (i.e. forcing the samples of the same class to be closer to each other) and maximizing the inter-class variance (i.e. pushing different classes away from each other) [21]. Ensuring both aspects facilitate the extraction of highly discriminative features [7]. This can have a significant effect on improving the performance of systems that require embeddings extraction, such as speaker diarization.

The choice of the loss function greatly affects the performance of Deep Neural Networks (DNNs) on effectively extracting feature embeddings from input data, where the extracted embeddings should be discriminative [26]. Discriminative means that the extracted embeddings corresponding to the same class should be close to each other in the embeddings space, while embeddings corresponding to different classes should be further apart from each other. Therefore, a successful loss function should be capable of minimizing the intra-class variance and maximizing the inter-class variance as much as possible to achieve discriminative embeddings [26].

During the training process of a CNN, which is a type of DNNs, the model learns to extract useful features from the training data via the loss function, such that the trained model can be used afterward as a feature (embeddings) extractor [27]. For example, in this thesis, the SincNet model [1], which is basically a CNN architecture, is trained using different state-of-the-art losses to perform speaker recognition, where the trained model will be used to extract discriminative speaker embeddings (d-vectors) in a speaker diarization system. The softmax loss is widely used to train CNNs, since it facilities the learning of separable features [27]. However, in cases where there is a chance that the variations between samples belonging to the same class can be larger than the variations between samples belonging to different classes, it is not enough to learn features that are separable, they must also be discriminative [27]. Therefore, the softmax loss will not be effective for such cases, which gave the motivation to design new loss functions that can facilitate the learning of discriminative features [27]. Generally, the losses that can effectively learn discriminative features can be categorized into two categories: metric-based losses and cosine-margin-based losses [27].

2.3.1 Metric-Based Losses

Firstly, the metric-based losses employ Euclidean distance to embed the input data into Euclidean space, where the intra-class variance is decreased and the inter-class variance is increased [27]. The contrastive loss [28] and the triplet loss [29] are widely used metric-based losses.

The contrastive loss takes input samples pairs, where it reduces the euclidean distance between pairs that belong to the same class, and increases the distance between pairs that belong to different classes [27]. Moreover, the work in [30], [31], and [32] combined the contrastive loss and the softmax loss to achieve better features extraction performance [27]. However, the major issue of the contrastive loss is the difficulty of selecting the margin parameters [27].

The triplet loss uses the relative distance between pairs as opposed to the contrastive loss which uses the absolute distance between pairs [27]. The triplet loss uses triplets of input samples, where it reduces the distance between an anchor sample and a sample from the same class, and it increases the distance between the anchor sample and a sample from another class [27]. Additionally, the work in [33], [34], [35], [36] uses both the softmax loss (for training) and the triplet loss (for fine-tuning) in their implementation [27].

However, the contrastive loss and the triplet loss can suffer from training instability because of the need for an effective sampling strategy [27]. Thus, the center loss [37], and its other forms [38], [39], [40], and [41] can be used as an alternative to decrease intra-class variance [27]. The center loss determines the center of every class, and then it reduces the distance between the learned features of every class and its center [27]. Nevertheless, the center loss and its other variations have huge GPU memory requirements, and they work best if the training data is balanced and has an adequate number of samples for each class [27].

2.3.2 Cosine-Margin-Based Losses

Metric loss functions are computationally expensive, and they require efficient mining methods, where the performance of the metric loss functions is highly affected by the mining method [42]. Thus, the majority of loss functions in the literature concentrates on modifying the conventional classification loss functions, which includes the softmax loss and its variants (mainly margin-based) to achieve high discriminative performance [42]. The network is first trained using the softmax loss, or one of its variants, to perform the classification task, then the trained network is used for features (embeddings) extraction.

The idea behind cosine (angular) margin-based losses is to ensure that learned features are strongly separated by imposing a larger cosine distance [27]. It is worth noting that the margin-based losses use the softmax loss in their implementation, where they modify it to incorporate stricter margins [27].

The large-margin softmax (L-Softmax) loss [43], introduces an angular margin to the conventional softmax loss by reformulating the softmax loss and introducing a margin parameter (m) [27]. The soft-margin softmax (SM-Softmax) loss [44] modified the L-Softmax loss to improve its ability to learn better discriminative features by replacing the hard angular margin with a soft distance margin, such that this will increase the intra-class compactness and the inter-class separability. The A-Softmax loss [45] improved on the L-Softmax loss by normalizing the weights of the classification layer (i.e. the weights used in the loss calculation) using L2 norm, where these normalized vectors will lie on the unit sphere [27]. Thus, an angular margin can be used to facilitate the learning of discriminative features on the unit sphere [27].

The L-Softmax and the A-Softmax loss functions are difficult to optimize, since they integrate the cosine margin in a multiplicative way [27]. Therefore, the AM-Softmax [21], CosFace [46], and ArcFace [22] were proposed to overcome this issue by incorporating the cosine margin in an additive manner [27]. These three losses have clear interpretation and easy implementation, while also having a better convergence compared to L-Softmax and the A-Softmax [27]. Moreover, as opposite to L-Softmax and the A-Softmax, the additive margin-based losses have a simpler formulation, and they do not have complicated hyperparameters [27]. Additionally, inspired by the NormFace loss [47], which studied the importance of normalizing the weights and features of the classification layer, the aforementioned additive margin losses apply the same normalization, since this facilities a better performance [27]. Some margin-based losses, such as FairLoss [48] and AdaptiveFace [49], where developed to handle cases of unbalanced data by adaptively changing the margins for the various classes [27].

Furthermore, due to the promising performance of the margin-based losses (especially additive margin losses), the literature has several other cases, where authors worked on modifying the margin-based losses to further enhance their discriminative performance. For example, multiple authors focused on improving the AM-Softmax loss. The double additive margin softmax (DAM-Softmax) modified the AM-Softmax loss [21] by incorporating an additive margin to both the intra-class similarity and inter-class similarity components of the loss, as opposite to the AM-Softmax loss, which only incorporate the margin to the intra-class similarity component of the loss [50]. This will lead to a better discriminative features extraction compared to the AM-Softmax loss [50]. Other losses that improved on the AM-Softmax loss are the Dynamic-AM-Softmax loss [51], which dynamically changes the margin of every sample in the training set to facilitate stronger intra-class compactness. The ensemble additive margin softmax (EAM-Softmax) loss [52] incorporates the Hilbert-Schmidt independence criterion (HSIC) [53] with the AM-Softmax loss to achieve better performance.

In addition, the inter-class angular margin (IAM) loss [54] focuses on increasing the inter-class separability by increasing the angular margin among different classes, through dynamically applying stronger penalization on inter-class angles with smaller values, as opposite to other famous softmax losses that uses fixed inter-class margins [54]. The IAM loss can be easily combined with the softmax loss and its other variations, such as AM-Softmax [21], where it will act as a regularization term to improve their performance by boosting their ability to learn better discriminative features [54].

2.4 The Chosen State-of-the-Art Loss Functions

The main goal of this thesis is to improve the loss function of the SincNet model [1], in order to improve its ability to extract discriminative embeddings, and then employ the SincNet with an improved loss in a speaker diarization system. As mentioned before, most of the literature focuses on improving the Softmax loss and its variants, such as the Additive Margin Softmax (AM-Softmax) loss [21], in order to achieve high recognition performance by enhancing the ability to extract discriminative embeddings. This is attributed to the superior performance of the variants of Softmax loss, and their advantages over metric-based loss functions, such as contrastive loss and triplet loss [42]. Four potential loss functions were chosen to be combined with the SincNet architecture [1]. The chosen loss functions are:

1. AdaCos, which is an adaptive hyperparameter-free cosine-based loss [55]

2. Pairwise Gaussian loss (PGL), which combines Softmax loss and contrastive loss in an efficient and effective way [56]

3. MV-Softmax, which has the advantages of margin-based Softmax losses and mining-based Softmax losses [42]

4. D-Softmax, which disentangles the intra-class objective and inter-class objective and optimize them independently [57]

These four loss functions were chosen because, in addition to their state-of-the-art performance, they are novel and innovative in their approach of ensuring a minimal intra-class variance and a maximal inter-class variance. Additionally, each of the four losses offers a highly unique solution to improve on the Softmax loss, as opposed to the majority of the other loss functions, such as [50], [51], [52], and [54], where they concentrate solely on improving the margin-based Softmax losses by mostly changing their formulation. The fact that the four chosen loss functions are highly unique from each other will facilitate a better comparison between them. This will give a better indicator to know which improvement on the Softmax loss will result in the best recognition performance and the best embeddings extraction performance.

In the following sections, I will first discuss the existing losses that have already been used with the SincNet architecture. Then, I will thoroughly overview the chosen state-of-the-art loss functions that will be combined with the SincNet based architectures.

2.4.1 Currently Combined Loss Functions with SincNet

The original SincNet model [1] employed the widely used softmax loss. The softmax loss focuses on maximizing the inter-class variance, resulting in separable features [26]. However, the softmax loss does not result in discriminative embeddings, since it does not try to minimize the intra-class variance [9]. This will deprive the original SincNet from extracting highly separable and discriminative speaker embeddings. Therefore, some authors improved the performance of SincNet by employing better loss functions.

For example, as mentioned in Section 2.2.2, the authors of [9] proposed AM-SincNet that used the Additive Margin Softmax (AM-Softmax) loss, which improves on the conventional Softmax loss by incorporating additive margin in the decision boundary [21], to train the SincNet model. The AM-SincNet offered a significant speaker recognition improvement over the conventional SincNet, since the AM-Softmax loss is capable of minimizing the distance between samples corresponding to the same class and maximizing the distance between different classes simultaneously by incorporating additive margin in the decision boundary [9].

Additionally, as mentioned in Section 2.2.2, the authors of [10] proposed AF-SincNet which employs Additive Angular Margin Softmax loss (also called ArcFace) [22], to improve the performance of the SincNet model. Both of AM-SincNet [9] and AF-SincNet [10] gave a superior performance over the SincNet model [1], which shows the significance of improving the loss function of the SincNet model.

2.4.2 The AdaCos Loss

The performance of the conventional Softmax loss was greatly improved by introducing the cosine-based with margin functions, such as L-Softmax [43], A-Softmax [45], and AM-Softmax [21], where these methods give a high classification performance [55]. These losses are especially useful in the cases of open-set problems, where the testing classes are not part of the training classes [55].

However, even though the cosine-based Softmax losses are successful, their performance is very sensitive to their hyperparameters settings [55]. Moreover, minor changes of the hyperparameters may result in the failure of the network's training [55]. The hyperparameters are chosen experimentally through a large number of trials, which is a time-consuming process [55]. There are algorithms that can change the hyperparameters of a given model automatically, such as Random search [58] and sequential model-based optimization [59], but the operation of these algorithms involve running several trials to find the best hyperparameters settings, which is time consuming [55]. Therefore, finding a way to adaptively tune the hyperparameters in cosine-based Softmax losses without the need for running numerous trials would be highly useful.

Before proposing the AdaCos loss function with the adaptive hyperparameters, the authors in [55] studied the effect of the two main hyperparameters of the cosine-based Softmax losses, which are the scaling (s) and the margin (m) parameters. The s parameter is responsible for making the different classes more discriminative by up scaling the cosine distances. However, the m parameter improves the classification performance by increasing the margin between the various classes. The authors of [55] discussed the effect of these parameters on the prediction probability, where they found that it is highly affected by both hyperparameters. Two important properties must be satisfied when choosing the best hyperparameters for margin-based softmax losses [55]. The first property is to ensure that the range of predicted probability for a given class covers the values between 0 and 1 [55]. The second property is to ensure that the gradient of the prediction probability of a given class should be big enough with respect to the angle between the input features to the classification layer and the weights of that given class, which will facilitate the training process [55]. Additionally, the authors of [55] found that the effect of both parameters can be unified, since a large s and a small m reduces the network's supervision. On the other hand, a small s and a large m enhances the network's supervision (i.e. the training becomes more strict, which decreases the intra-class variance). Thus, it is possible to focus on one of these two hyperparameters to control the behavior of the training [55]. The authors of the AdaCos loss [55] chose to concentrate on auto-tuning the s parameter, because it greatly affects the range and the shape of the predicted probabilities' curves, which greatly influences the training performance of the network, as opposed to the m parameter, which only shifts the curves of the predicted probabilities left and right.

Therefore, the authors of [55] proposed the AdaCos loss function, where they removed the m parameter, and they designed an automatically changing scaling parameter s. Thus, the AdaCos loss overcomes the time-intensive manual hyperparameters tuning. The authors of [55] proposed two ways to automatically choosing the s parameter of the AdaCos loss: (1) setting the s parameter to a fixed value (2) dynamically changing the s parameter.

In [55], the authors derived a fixed value that can be used to set the s parameter of the AdaCos loss, where they showed that the s parameter can be set to:

$$\tilde{s_f} = \sqrt{2} \cdot \log\left(C - 1\right) \tag{2.9}$$

where $\tilde{s_f}$ is the fixed scale parameter, and C is the number of classes in the training dataset. This derived fixed s parameter value will be used mainly as a baseline to compare its performance to the dynamically changing the s parameter [55]. Moreover, $\tilde{s_f}$ can serve as good choice for the s parameter for other margin-based losses, instead of manually changing s parameter for these losses [55].

Additionally, the authors in [55] derived a dynamically changing s parameter for the AdaCos loss, where as opposite to $\tilde{s_f}$, it can facilitate a great supervision during training, by gradually applying stricter requirements on the angle between the weights of a certain class and the input features to the classification layer as the training progresses [55]. From [55], the dynamically changing s parameter ($\tilde{s_d}$) is calculated as:

$$\tilde{s_d}^{(t)} = \begin{cases} \sqrt{2} \cdot \log\left(C - 1\right) & t = 0, \\ \frac{\log B_{avg}^{(t)}}{\cos\left(min(\frac{\pi}{4}, \theta_{med}^{(t)})\right)} & t \geq 1, \end{cases} \tag{2.10}$$

where $\tilde{s_d}^{(t)}$ is the adaptive scale parameter at the t^{th} epoch, $\theta_{med}^{(t)}$, which is found during each mini-batch of size N at epoch t, is the median of all the angles between the feature vector x_i and the weight vector W_{y_i} corresponding to its correct class y_i. $B_{avg}^{(t)}$ is calculated as:

$$B_{avg}^{(t)} = \frac{1}{N} \sum_{i \in N^{(t)}} \sum_{k \neq y_i} e^{\tilde{s_d}^{(t-1)} \cdot \cos\theta_{i,k}} \tag{2.11}$$

From Eq. 2.10,It can be seen that at $t = 0$, the value of $\tilde{s_d}$ is equal to $\tilde{s_f}$, after that $\tilde{s_d}$ will start dynamically changing. The AdaCos loss using the adaptive s parameter can be written as:

$$L_{AdaCos} = -\frac{1}{N}\sum_{i=1}^{N} \log P_{i,y_i} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\tilde{s_d}^{(t)} \cdot \cos\theta_{i,y_i}}}{\sum_{k=1}^{C} e^{\tilde{s_d}^{(t)} \cdot \cos\theta_{i,k}}} \tag{2.12}$$

where P_{i,y_i} is the predicted probability of the correct class y_i for the input sample i, $\theta_{i,j}$ is the angle between the x_i feature vector corresponding to input sample i, and the weights of the j^{th} class W_j, and $\cos\theta_{i,j}$ is:

$$\cos\theta_{i,j} = \frac{\langle \vec{x_i}, \vec{W_j} \rangle}{\|\vec{x_i}\| \|\vec{W_j}\|} \tag{2.13}$$

In [55], the authors used the AdaCos loss with two different architectures, which are the ResNet-50 [23] and Inception-ResNet [60] to perform face recognition on several widely used datasets. They compared the performance of the AdaCos loss with other margin-based softmax losses, such as ArcFace [22]. Their results showed the effectiveness of the AdaCos loss, where it gave the best recognition performance compared to the other loss functions, especially when using the dynamic s parameter. Moreover, the AdaCos had the best convergence rates compared to the other margin-based softmax losses [55].

2.4.3 The Pairwise Gaussian Loss (PGL)

Loss functions should facilitate the learning of robust features from the input data, which will help in improving the classification performance of Convolutional Neural Networks (CNNs) [56]. Strong learned features are characterized by an adequately maximized intra-class compactness and inter-class separability [56]. The softmax loss, which combines the softmax activation function with the cross-entropy loss function, is widely used with CNNs to perform classification tasks [56]. Although the softmax loss is capable of maximizing the inter-class separability, it lacks the ability to maximize the intra-class compactness, which will negatively affect the ability of the CNN to learn discriminative features [56].

Thus, several improvements on the softmax loss were proposed, in which stricter constraints were imposed on the softmax loss to improve its performance [56], such as the additive margin softmax (AM-Softmax) [21], and the ArcFace loss [22]. Even though these losses give an improved classification performance over the conventional softmax loss, a more desirable option would be to directly maximize the intra-class compactness [56]. To illustrate, it is highly useful to employ a loss function that can directly optimize the intra-class compactness, and then combine that loss with the conventional softmax loss, which is capable of optimizing the inter-class separability [56]. There are different losses that can directly optimize the intra-class compactness, such as the contrastive loss [28] and the triplet loss [29], where the contrastive loss is the best candidate compared to the other methods, because it overcome the drawbacks of the other options, such as convergence issues and high time cost [56]. The contrastive loss calculates the euclidean distance between a pair of feature vectors. However, in order to effectively combine the contrastive loss with the softmax loss, they should have the same range, which is between 0 and 1, since the output of the softmax loss is a probability distribution [56]. There are different methods to map the distance measure of the contrastive loss into a similarity measure, such as Hinge-like [61] and Sigmoid [62], but these methods have several shortcomings, including the use of incompatible similarity measures with the softmax loss, and convergence issues [56].

Therefore, the authors of [56] proposed the Pairwise Gaussian Loss (PGL), which directly optimizes the intra-class compactness by greatly penalizing sample pairs that belong to the same class but are far away from others, to overcome the previously mentioned shortcomings of the different mapping functions. PGL employs a Gaussian function, which can result in a fast model convergence, to map the Euclidean distance of the contrastive loss into a similarity measure that is between 0 and 1, which facilitate the combination of PGL and softmax loss [56]. The Euclidean distance is calculated using the last hidden fully connected layer (i.e. the penultimate layer) of a CNN between a pair of samples in a given batch. PGL employs cross-entropy loss, and it will be combined with the softmax loss to create the total loss that will be used to train the network.

Additionally, the contrastive loss requires a complex sample organization step, which is time-consuming [56]. Thus, the authors in [56] overcame this issue by taking advantage of the fact that the input data to the CNN is random [56]. Given that the number of randomly chosen samples in a mini-batch during training is even, the authors split the mini-batch into an odd group and an even group depending on the samples' indices. Then, the pairwise distance is calculated using the penultimate layer between each sample from the odd group with its corresponding sample from the even group, where the label of the pairwise distance is 1 if the pair samples belong to the same class, and 0 otherwise.

The major part of PGL is the Gaussian function that is used to map the distance measure of the contrastive loss into a similarity measure [56]. The authors of [56] defined their Gaussian mapping function as:

$$g(d_{ij}) = e^{-\beta d_{ij}^2} \tag{2.14}$$

where, d_{ij} is the Euclidean distance between feature vectors i and j, β is a scale parameter.

Thus, the probability mapping function is:

$$\begin{aligned} p(y_{ij}|d_{ij}) &= \begin{cases} g(d_{ij}) & y_{ij} = 1 \\ 1 - g(d_{ij}) & y_{ij} = 1 \end{cases} \\ &= [g(d_{ij})]^{y_{ij}}[1 - g(d_{ij})]^{1-y_{ij}} \end{aligned} \tag{2.15}$$

where $y_{ij} = 1$ means that the feature vectors i and j correspond to the same class, while $y_{ij} = 0$ means that they are not from the same class. Therefore, the PGL can be written as follows:

$$\begin{aligned} L_{PGL} &= \frac{2}{N} \sum_{i \in odd}^{N} \sum_{j \in even}^{N} -\log(p(y_{ij}|d_{ij})) \\ &= \frac{2}{N} \sum_{i \in odd}^{N} \sum_{j \in even}^{N} [\beta d_{ij}^2 + (y_{ij} - 1)\log(e^{\beta d_{ij}^2} -)] \end{aligned} \tag{2.16}$$

Finally, the total loss that was used to train the CNNs in [56], would be:

$$L_{total} = L_{softmax} + L_{PGL} \tag{2.17}$$

$L_{softmax}$ (the softmax loss) will concentrate on maximizing the inter-class separability, while L_{PGL} will concentrate on maximizing the intra-class compactness.

The authors of PGL [56], combined their loss with several CNN architectures, mainly VGG-net [63] and ResNet-50 [23], and they tested there models on multiple famous image recognition datasets. The authors compared PGL with various other losses, and different mapping functions for the contrastive loss. Their results showed the high performance of the PGL, where it gave the best recognition performance. Additionally, the PGL had a fast convergence rate compared to the other losses it got compared with, and the authors showed its ability to be effectively combined with different CNN architectures [56].

2.4.4 The MV-Softmax Loss

The existing margin-based softmax losses, such as the AM-Softmax [21], are capable of achieving high recognition performance, but they do not take advantage of feature mining techniques, which are useful for learning discriminative features [42]. Moreover, they focus on increasing the margins with respect to the ground-truth class without considering the other classes, and they use the same fixed margin between all classes, which may not be practical [42].

The existing mining-based softmax losses, such as HM-Softmax [64] and F-Softmax [65], aims to enhance the capability to learn discriminative features by concentrating on hard examples (i.e. samples that are hard to classify) [42]. However, these mining-based softmax methods suffer from multiple drawbacks that can hinder their performance. For example, the HM-Softmax loss [64], which is a hard mining method that tries to improve features learning by using samples with high loss to create mini-batches for training, fully ignores the easy samples, and the number of hard samples is found experimentally [42]. On the other hand, the F-Softmax loss [65] uses a softer mining technique compared to HM-Softmax [64], where it uses a sparse set of hard samples for its training, but it does not clearly indicate hard samples [42]. Both HM-Softmax [64] and F-Softmax [65] do not semantically choose the hard samples [42].

Margin based loss functions aim to increase the margin between different classes, whereas the goal of mining-based loss functions is to concentrate on the hard examples (miss-classified samples) [42]. This means that margin-based and mining-based methods are independent from each others, and can be easily combined [42]. Thus, the authors of [42] proposed the MV-Softmax loss which is a margin-based softmax loss that can adaptively focus on the incorrectly classified samples to facilitate the learning of discriminative features, which will solve the issues mentioned in the previous paragraphs in order to achieve high recognition performance. The MV-Softmax loss is the first loss to combine the benefits of margin-based and feature mining methods in a single loss [42]. The MV-Softmax loss is defined as:

$$L_{MV} = -\log \frac{e^{sf(m,\theta_{w_y,x})}}{e^{sf(m,\theta_{w_y,x})} + \sum_{k\neq y}^{K} h(t,\theta_{w_k,x},I_k)e^{s\cos(\theta_{w_k,x})}} \tag{2.18}$$

To better understand the equation of the MV-Softmax loss [42] (Eq. 2.18), let us discuss its components separately.

$f(m, \theta_{w_y,x})$ is a margin function that can be either A-Softmax loss [45], AM-Softmax loss [21], or Arc-Softmax loss [22], where the authors of [42] combined the three mentioned softmax based losses in a single expression that is:

$$f(m, \theta_{w_y,x}) = \cos(m_1\theta_{w_y,x} + m_3) - m_2 \tag{2.19}$$

where $\theta_{w_y,x}$ is the angle between the weight vector of the ground-truth class y, and vector x, which is the input vector to the classification layer, $m_1 \geq 1$ is an integer that corresponds to the A-Softmax loss [45], m_2 is a decimal number that is between 0 and 1, and it corresponds to the AM-Softmax loss [21], m_3 is a decimal number that is between 0 and 1, and it corresponds to the Arc-Softmax loss [22]. To illustrate, if the AM-Softmax loss [21] is used, the margin function will be $f(m_2, \theta_{w_y,x}) = \cos(\theta_{w_y,x}) - m_2$.

$h(t, \theta_{w_k,x}, I_k) \geq 1$ is a re-weighting function that is used to focus on the miss-classified samples. In [42], the authors proposed two formulations for their re-weighting function. The first uses the same fixed weights for all of the wrongly classified classes, and it is defined as:

$$h(t, \theta_{w_k,x}, I_k) = e^{stI_k} \tag{2.20}$$

The second uses adaptive weights for the wrongly classified classes, and it is defined as:

$$h(t, \theta_{w_k,x}, I_k) = e^{st(\cos(\theta_{w_k,x})+1)I_k} \tag{2.21}$$

where s is a scale parameter that authors of [42] fixed it to 32, $t \geq 0$ is preset hyperparameter, and I_k is a binary indicator function that can adaptively specify if a sample got miss-classified by a given classifier w_k, where $k \neq y$, and it is defined as:

$$I_k = \begin{cases} 0, & f(m, \theta_{w_y,x}) - \cos(\theta_{w_k,x}) \geq 0 \\ 1, & f(m, \theta_{w_y,x}) - \cos(\theta_{w_k,x}) < 0 \end{cases} \tag{2.22}$$

The indicator function I_k will ensure that the MV-Softmax loss [42] can explicitly focus on hard examples (miss-classified samples) to enhance the ability to learn discriminative features, since hard samples can play a greater role in improving the ability of loss to learn discriminative features compared to easily classified samples [42].

The authors of the MV-Softmax loss [42], which inherits the advantages of margin-based and mining-based methods, used two variations of the MV-Softmax loss in their experiments. They used the MV-AM-Softmax loss, which uses the AM-Softmax loss [21] as the margin function in the MV-Softmax loss, and MV-Arc-Softmax loss, which uses the Arc-Softmax loss [22] as the margin function in the MV-Softmax loss. The authors of [42] designed a CNN architecture called AttentionNet-IRSE, which merges AttentionNet [66] with an IRSE module [22], where they combined it with the MV-Softmax loss. In [42], the authors applied their models on the face recognition task, where they tested their models using different widely used datasets. The authors of [42] compared their MV-AM-Softmax and MV-Arc-Softmax losses with several other margin-based losses, such as AM-Softmax [21], and mining-based losses, such as HM-Softmax [64], where both MV-AM-Softmax and MV-Arc-Softmax outperformed the other losses. In addition, the authors of [42] found that using an adaptive re-weighting function instead of a fixed one in Eq. 2.18, results in a better performance, since the adaptive re-weighting function gives more weight to the harder samples, which enhances the ability to learn better discriminative features.

2.4.5 The D-Softmax Loss

The widely used softmax loss has its intra-class objective and inter-class objective entangled, which means that focusing on optimizing one of these objectives, will cause a relaxation of the other [57].

Many existing methods, such as Cosface loss [46], mainly focused on addressing the weaknesses of the softmax loss by improving the constraints through introducing large margins to the softmax loss, and they ignored the fact that lack in performance of the softmax loss is the byproduct of the interconnected intra-class objective and inter-class objective [57]. Therefore, the authors of [57], who were the first to address the entanglement issue of the softmax loss, proposed the D-Softmax loss which disentangles the conventional softmax loss into two independent objectives, the inter-class objective and the intra-class objective, where they can be optimized separately. The intra-class objective will focus on compacting the embedding vectors corresponding to the same class in the embeddings space until a given condition is met [57]. The inter-class objective will focus on keeping the embeddings vectors corresponding to different classes far away from each other in the embedding space [57].

The authors of [57] defined the intra-class objective of the D-Softmax loss as:

$$L_D^{intra} = \log\left(1 + \frac{\epsilon}{e^{sz_y}}\right) \tag{2.23}$$

where s is a scale parameter, which was fixed to 32 by the authors of [57], z_y is the activation of the ground-truth class that is defined as $z_y = \cos(\theta_{w_y,x})$, where $\theta_{w_y,x}$ is the angle between the weights of the ground-truth class and the input vector to the classification layer. ϵ is a constant value that was defined by the authors of [57] as:

$$\epsilon = e^{s \cdot d} \tag{2.24}$$

where d acts as a termination point to the optimization process to avoid vanishing gradients [57]. The value of d can be manually adjusted such that it has a large enough value, which will ensure that the intra-class objective is well optimized [57]. Since, the authors of [57] found that the softmax loss causes the weights of the different classes to be largely separated, which causes the early termination of the intra-class objective. This hinders the ability of the softmax loss to learn compact features that belong to the same class [57]. Also, it was found that the margin-based losses, such as ArcFace [22], has a sufficiently large optimization termination point compared to the softmax loss, which explains the superior performance of the margin-based losses [57]. This is what inspired the authors of

[57] to introduce the previously mentioned d parameter, and setting it to a large enough value, in order to overcome the early termination problem of the softmax loss.

After defining the intra-class component of the D-Softmax loss (seen in Eq. 2.23), the authors of [57], defined the inter-class component of the D-Softmax loss, which works as a regularization tool to prevent the D-Softmax loss from reaching a simple solution, which maps all the samples to a single point. The authors of [57] defined the inter-class objective as:

$$L_D^{inter} = \log\left(1 + \sum_{k \neq y} e^{sz_k}\right) \tag{2.25}$$

where s is a scale parameter, k is the k-th class ($k \in 1, 2, ..., K$ and $K = Number\ of\ classes$)), y is the ground-truth class, z_k is the k-th class activation, and it is defined as $z_k = \cos(\theta_{w_k,x})$, where $\theta_{w_k,x}$ is the angle between the weights of the k-th class and the input vector to the classification layer. The formulation of the inter-class objective in Eq. 2.25 ensures strict regularization on its optimization [57]. Therefore, using Eq. 2.23 and Eq. 2.25, the D-Softmax loss is defined as:

$$L_D = L_D^{intra} + L_D^{inter} \tag{2.26}$$

It is worth noting that the authors of [57] addressed the fact that the softmax loss and its modified versions suffer from high time and memory requirements, since they involve the activations of all the classes in their calculation. One of the properties of the D-Softmax loss is that it splits the computational requirements into two separate part [57]. The intra-class objective is not computationally expensive, since it uses the activation of the ground-truth class only, but the inter-class objective has a high computational cost because it uses the activations of all the non-ground-truth classes [57]. The authors of the D-SincNet loss [57] found that the high computational cost of the inter-class objective is unnecessary, and it can be greatly reduced by sampling a subset of the non-ground-truth classes for the calculation of the inter-class objective. Thus, the authors of [57] proposed two additional D-Softmax based losses, which are the D-Softmax-B and D-Softmax-K, where they use two distinct sampling techniques. Both D-Softmax-B and D-Softmax-K were able to largely reduce the time cost of training, while barely reducing the recognition performance [57]. However, in this thesis, only the normal D-Softmax loss will be used, since the accelerated versions of it are out of the scope of this thesis.

The authors of the D-Softmax loss [57] used their loss with the ResNet-50 architecture [23], and they applied their models on the face recognition task. In [57], the authors used several widely used face recognition datasets to assess their model performance. In addition, they compared the D-Softmax loss with different margin-based softmax losses, such as ArcFace [22], where the D-Softmax outperformed the other losses.

2.4.6 Summary of the Chosen Loss Functions

After reviewing various loss functions, it was found the following four state-of-the-art loss functions to be the most novel and innovative in their approach to enhance the performance of the Softmax loss. Therefore, the following loss functions will be considered to enhance the performance of the SincNet based architectures, and then their performance improvement will be compared.

The first loss function is the AdaCos, which is an adaptive cosine-based Softmax loss that does not need manual hyperparameters tuning [55]. Standard cosine-based Softmax loss functions, such as AM-Softmax [21] and ArcFace [22], are capable of achieving a great classification performance [55]. However, these loss functions require manual hyperparameters tuning through multiple trails, which is time-consuming and may result in sub-optimal performance [55]. Thus, the authors of [55] thoroughly analysed two major hyperparameters that affect the performance of cosine-based loss functions, which are the scale and the angular margin parameters [55]. Then, they used their analysis to come up with the AdaCos loss function, which performs automatic hyperparameters tuning [55]. The AdaCos loss function was proposed to enhance the performance of the face recognition task, where it gave a superior performance [55].

The second loss is the pairwise Gaussian loss (PGL), which focuses on minimizing the intra-class variance by greatly penalizing sample pairs that correspond to the same class but are far away from each other in the embedding space [56]. The Softmax loss focuses only on maximizing the inter-class separability. Therefore, combining the Softmax loss with a loss function that focuses on minimizing the intra-class variance, such as the contrastive loss [28] , will result in an optimum classification performance [56]. Contrastive loss is a metric-based loss that can be used to directly optimize the intra-class compactness by measuring the distance between pairs of samples [56]. However, in order to effectively combine the Softmax loss and the contrastive loss, a proper mapping must be used to map the Euclidean distance that is calculated using the contrastive loss into a range that suits the Softmax loss (i.e. between 0 and 1) [56]. Moreover, the contrastive loss requires a complex sample mining procedure [56]. Thus, the authors in [56] solved both of the aforementioned issues by mapping the Euclidean distance between pairs of samples into the range [0,1] using a Gaussian function [56]. Moreover, they solved the samples mining problem by exploiting a simple yet effective pairwise feature vectors organization, which is done at the penultimate layer [56]. The PGL loss ensures great classification performance, fast convergence, and high generalizability [56]

The third loss is the MV-Softmax loss which concentrates on the miss-classified feature vectors to guide the learning process in an adaptive manner [42]. The MV-Softmax loss

[42] came as a solution to tackle the drawbacks of margin-based loss functions, such as AM-Softmax [21]. For example, the margin-based losses do not take into consideration the feature mining process, which is important to achieve highly discriminative embeddings [42]. On the other hand, mining-based softmax losses, such as HM-Softmax [64], also suffer from some problems that can affect their performance negatively [42]. The MV-Softmax loss is a novel approach, where it is the first loss function to combine the advantages of feature mining and feature margin methods [42]. The MV-Softmax loss was tested on the face recognition task, where it outperformed mining-based Softmax loss functions and the margin-based Softmax loss functions [42].

Lastly, the fourth loss function is the D-Softmax loss, which split the conventional Softmax loss function into an intra-class variance part and an inter-class variance part, where each part can be optimized independently [57]. Most of the existing methods that improve on the discriminative ability of conventional Softmax, such as incorporating angular margin in the Softmax formulation, focus on strengthening the constraints, while ignoring the fact that the inadequacy of the conventional Softmax is due to the entanglement of the intra-class objective and the inter-class objective [57]. The authors of [57] were the first to explicitly address the problem of intra-class objective and intra-class objective entanglement present in the conventional Softmax loss. To illustrate, optimizing the inter-class variance in the Softmax loss will result in relaxing the intra-class variance optimization requirements [57]. Therefore, the authors of [57] proposed the dissection (or disentanglement) of the Softmax loss into two independent objectives, the intra-class objective and the inter-class objective, where they can be optimized separately. The D-Softmax loss was tested on the face recognition task, where it gave a superior recognition performance [57].

Chapter 3

The Proposed SincNet Models

In this chapter, I will go through the proposed SincNet models, where different SincNet based architectures are being combined with distinct state-of-the-art loss functions. Firstly, I will introduce four proposed models that combine the SincNet architecture [1] with four different loss functions. Secondly, I will go over six proposed models that combine the Res-SincNet architecture [2] with six different loss functions. Finally, I will go through the proposed Res-SincNet-FC architecture, where it will be also combined with six different loss functions.

3.1 Models Based on the SincNet Architecture

In this section, four different models that combine the SincNet architecture [1] with four different state-of-the-art loss functions will be proposed. The general SincNet architecture can be seen in Fig. 3.1. It is important to note that all the operations involved in SincNet, such as convolution and pooling, are 1-Dimensional.

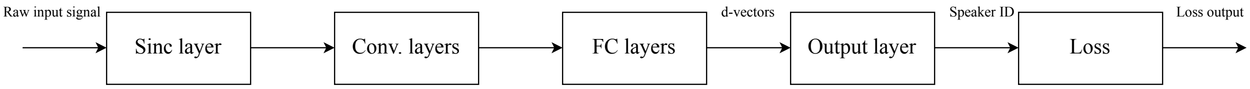

Figure 3.1: The main components of SincNet architecture [1]

From Fig. 3.1, it can be seen that the SincNet architecture has the following main components:

- Sinc layer: This layer consists of parameterized sinc filters, and it acts as the input to the SincNet model, where it can process raw input waveforms efficiently and effectively. This layer has 80 filters of length 251 samples (unless stated otherwise), and a stride of one sample. Moreover, after the convolution operation, this layer applies max-pooling using filters of length three samples, followed by layer normalization [67] (it was also applied to the input signal), followed by leaky ReLU activation function [68].
- Conv. layers: The sinc layer is followed by two standard convolutional layers, each having 60 filters of length five samples and a stride of one sample. Both convolutional layers employ max-pooling using filters of length three samples, followed by layer normalization [67], followed by leaky ReLU activation function [68].
- FC layers: The convolutional layers are followed by three fully connected layers, each having 2048 neurons, and all of them employ batch normalization [69], and leaky ReLU activation function [68]. The speaker embeddings (d-vectors) are extracted from the output of the last fully connected layer.
- Output layer: The output layer is the last layer in the network, and it has the same number of neurons as the number of speakers in the training dataset. The output of this layer is the predicted speaker ID.
- Loss: The loss function that will be used to train the SincNet model.

All of the aforementioned hyperparameter choices for the SincNet architecture are the ones used in its original paper [1]. For each of the following models that are based on the SincNet architecture [1], the same architecture seen in 3.1 is used, but the loss function is changed.

3.1.1 AdaCos-SincNet

The first proposed model is the AdaCos-SincNet model. The proposed AdaCos-SincNet model combines the SincNet architecture [1] (depicted in Fig. 3.1) with the AdaCos loss [55], to benefit from the AdaCos characteristics. The main advantage of the AdaCos loss [55] is that it does not require manual hyperparameter tuning, which is time consuming and may not result in an optimum performance, since it uses an adaptively changing parameter. The use of an adaptive hyperparameter will save the time needed to run many trials, and it will result in an improved optimization [55]. Moreover, the AdaCos loss will result in better performance and faster convergence [55].

3.1.2 PGL-SincNet

The proposed PGL-SincNet combines the SincNet architecture [1] (depicted in Fig. 3.1) with the Pairwise Gaussian Loss (PGL) [56], to take advantage of the PGL properties. PGL can effectively maximize intra-class compactness, since it can directly optimize it [56]. Thus, the weakness of the softmax loss to maximize intra-class compactness can be overcame by combining it with PGL [56]. PGL has less time cost and computational requirements compared to other similar methods, and it can achieve high performance with fast convergence [56].

3.1.3 MV-SincNet

The proposed MV-SincNet combines the SincNet architecture [1] (depicted in Fig. 3.1) with the MV-Softmax loss [42], to exploit the merits of the MV-Softmax loss. The MV-Softmax loss was the first to combine the advantages of margin based and mining based losses into a single loss [42], where it focuses on learning from miss-classified samples, which are more important to learn better discriminative features [42]. The MV-Softmax loss clearly indicates the hard samples and adaptively uses them to enhance the ability to learn highly discriminative features in order to improve the performance [42]. The MV-Softmax loss [42] has two versions, the MV-AM-Softmax loss, which uses the same margin of the AM-Softmax loss [21] with the MV-Softmax loss, and MV-Arc-softmax loss, which uses the same margin of the ArcFace loss [22] with the MV-Softmax loss. Therefore, two models are proposed, the MV-AM-SincNet and the MV-Arc-SincNet, which combines the SincNet architecture [1] (depicted in Fig. 3.1) with the MV-AM-Softmax loss and the MV-Arc-Softmax loss, respectively.

3.1.4 D-SincNet

The proposed D-SincNet combines the SincNet architecture [1] (depicted in Fig. 3.1) with the D-Softmax loss [57], to utilize the attributes of the D-Softmax loss. The D-Softmax loss [57] was the first loss to disentangle the softmax loss into intra-class objective and inter-class objective. The intra-class objective and the inter-class objective can be optimized independently, which will ensure that the constraints are always strict [57]. This will facilitate the learning of discriminative features, and thus enhancing the model performance [57].

3.2 Models Based on the Res-SincNet Architecture

In this section, different models that combine the Res-SincNet architecture [2] with six different state-of-the-art loss functions will be proposed. The general Res-SincNet architecture can be seen in Fig. 3.2. It is important to note that all the operations involved in Res-SincNet, such as convolution and pooling, are 1-Dimensional.

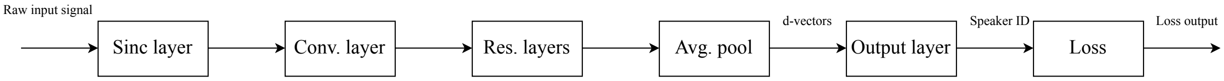

Figure 3.2: The main components of Res-SincNet architecture [2]

From Fig. 3.2, it can be seen that the Res-SincNet architecture has the following main components:

- Sinc layer: This layer consists of parameterized sinc filters, and it acts as the input to the SincNet model, where it can process raw input waveforms efficiently and effectively. This layer has 122 filters of length 89 samples (unless stated otherwise), a stride of one sample, and zero-padding of length 44 samples on both sides, which will maintain the input length. It is worth noting that the input signal is standardized (rescaling the data such that it has zero mean and unit variance) before passing it through the sinc layer.

- Conv. layer: The sinc layer is followed by a standard convolutional layer, having 512 filters of length nine samples, a stride of three samples, and zero-padding of length four samples on both sides. Also, the convolutional layer employs batch normalization [69], followed by ReLU activation function, followed by max-pooling using filters of length three samples, a stride of two samples, and zero-padding of size one sample on both sides.

- Res. layers: The convolutional layer is followed by four residual layers, the first residual layer consists of three residual blocks with 64 filters, the second residual layer consists of two residual blocks with 128 filters, the third residual layer consists of a single residual block with 256 filters, the last residual layer consists of a single residual layer with 512 filters. All the residual layers, except the first one, use a stride of 2, which will reduce the dimensionality by half after each residual layer. Also, they employ batch normalization [69] and ReLU activation.

- Avg. pool: The residual layers are followed by a global average pooling layer that will output 512 features. The speaker embeddings (d-vectors) are extracted from the output of this layer.

- Output layer: The output layer is the last layer in the network, and it has the same number of neurons as the number of speakers in the training dataset. The output of this layer is the predicted speaker ID.

- Loss: The loss function that will be used to train the Res-SincNet model.

All of the aforementioned hyperparameter choices for the Res-SincNet architecture are the optimal choices found in its original paper [2]. For each of the following models that are based on the Res-SincNet architecture [2], the same architecture seen in 3.2 is used, but the loss function is changed.

3.2.1 AM-Res-SincNet

The proposed AM-Res-SincNet combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the AM-Softmax [21], to benefit from the AM-Softmax loss characteristics. The AM-Softmax loss [21] facilitates the learning of highly discriminative features by imposing an additive margin to increase the angular distance between the different classes, which increases both intra-class compactness and inter-class separability.

3.2.2 Arc-Res-SincNet

The proposed Arc-Res-SincNet combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the ArcFace [22], to take advantage of the ArcFace loss properties. The ArcFace loss [22] is capable of effectively learning highly discriminative features due to the imposed angular margin, which encourages stronger intra-class compactness and inter-class separability.

3.2.3 AdaCos-Res-SincNet

The proposed AdaCos-Res-SincNet model combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the AdaCos loss [55], to benefit from the AdaCos characteristics, which were mentioned in Section 3.1.1.

3.2.4 PGL-Res-SincNet

The proposed PGL-Res-SincNet combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the Pairwise Gaussian Loss (PGL) [56], to take advantage of the PGL properties, which were mentioned in Section 3.1.2.

3.2.5 MV-Res-SincNet

The proposed MV-Res-SincNet combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the MV-Softmax loss [42], to exploit the merits of the MV-Softmax loss, which were mentioned in Section 3.1.3. The MV-Softmax loss [42] has two versions, the MV-AM-Softmax loss, which uses the same margin of the AM-Softmax loss [21] with the MV-Softmax loss, and MV-Arc-softmax loss, which uses the same margin of the ArcFace loss [22] with the MV-Softmax loss. Therefore, two models are proposed, the MV-AM-Res-SincNet and the MV-Arc-Res-SincNet, which combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the MV-AM-Softmax loss and the MV-Arc-Softmax loss, respectively.

3.2.6 D-Res-SincNet

The proposed D-Res-SincNet combines the Res-SincNet architecture [2] (depicted in Fig. 3.2) with the D-Softmax loss [57], to utilize the attributes of the D-Softmax loss, which were mentioned in Section 3.1.4.

3.3 Models Based on the Res-SincNet-FC Architecture

In this section, the proposed Res-SincNet-FC architecture will be used. The Res-SincNet-FC architecture has the same network components as the Res-SincNet architecture [2], except for the global average pooling layer, which was replaced with a fully connected layer. The use of a fully connected layer, instead of a global average pooling layer, is more suitable to be combined with the advanced softmax losses discussed in this thesis since it offers more flexibility. To elaborate, the fully connected layer will allow the losses to effectively learn discriminative features, that can be later extracted from its output as highly discriminative speaker embeddings (d-vectors).

In this section, different models that combine the Res-SincNet-FC architecture with six different state-of-the-art loss functions will be proposed. The general Res-SincNet-FC architecture can be seen in Fig. 3.3. It is important to note that all the operations involved in Res-SincNet-FC, such as convolution and pooling, are 1-Dimensional.

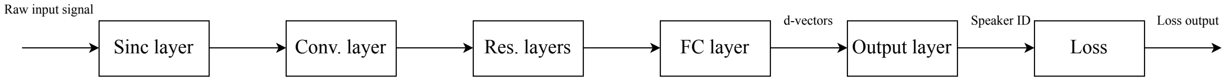

Figure 3.3: The main components of Res-SincNet-FC architecture

From Fig. 3.3, it can be seen that the Res-SincNet-FC architecture has the following main components:

- Sinc layer: This layer consists of parameterized sinc filters, and it acts as the input to the SincNet model, where it can process raw input waveforms efficiently and effectively. This layer has 122 filters of length 89 samples (unless stated otherwise), a stride of one sample, and zero-padding of length 44 samples on both sides, which will maintain the input length. It is worth noting that the input signal is standardized (rescaling the data such that it has zero mean and unit variance) before passing it through the sinc layer.

- Conv. layer: The sinc layer is followed by a standard convolutional layer, having 512 filters of length nine samples, a stride of three samples, and zero-padding of length four samples on both sides. Also, the convolutional layer employs batch normalization [69], followed by ReLU activation function, followed by max-pooling using filters of length three samples, a stride of two samples, and zero-padding of size one sample on both sides.

- Res. layers: The convolutional layer is followed by four residual layers, the first residual layer consists of three residual blocks with 64 filters, the second residual layer consists of two residual blocks with 128 filters, the third residual layer consists of a single residual block with 256 filters, the last residual layer consists of a single residual layer with 512 filters. All the residual layers, except the first one, use a stride of 2, which will reduce the dimensionality by half after each residual layer. Also, they employ batch normalization [69] and ReLU activation.

- FC layer: The residual layers are followed by the following sequence of layers: batch normalization (BN) [69], dropout [70] with a probability of 0.5, fully connected (FC) with 512 neurons, batch normalization (BN) [69], and leaky ReLU activation function [68]. The speaker embeddings (d-vectors) are extracted from the output of this layer.

- Output layer: The output layer is the last layer in the network, and it has the same number of neurons as the number of speakers in the training dataset. The output of this layer is the predicted speaker ID.
- Loss: The loss function that will be used to train the Res-SincNet-FC model.

For the mutual components between the Res-SincNet-FC architecture and the Res-SincNet architecture, the same hyperparameter choices as in the Res-SincNet architecture are used, which were the optimal choices found in the Res-SincNet architecture paper [2]. For the FC layer, experiments using different variations and combinations of the FC layer with other types of layers and activations was conducted, and it was found that the following combination of layers BN-dropout-FC-BN-leaky ReLU gives the best performance. It is worth noting that the replacement of the global average pooling layer with BN-dropout-FC-BN-leaky ReLU layers is inspired by the architecture used in the ArcFace paper [22], where they replaced the global average pooling layer in the ResNet architecture [23] with BN-dropout-FC-BN layers. For each of the following models that are based on the Res-SincNet architecture [2], the same architecture seen in 3.3 is used, but the loss function is changed.

3.3.1 AM-Res-SincNet-FC

The proposed AM-Res-SincNet-FC combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the AM-Softmax [21], to benefit from the AM-Softmax loss characteristics, which were mentioned in Section 3.2.1.

3.3.2 Arc-Res-SincNet-FC

The proposed Arc-Res-SincNet-FC combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the ArcFace [22], to take advantage of the ArcFace loss properties, which were mentioned in Section 3.2.2.

3.3.3 AdaCos-Res-SincNet-FC

The proposed AdaCos-Res-SincNet-FC model combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the AdaCos loss [55], to benefit from the AdaCos characteristics, which were mentioned in Section 3.1.1.

3.3.4 PGL-Res-SincNet-FC

The proposed PGL-Res-SincNet-FC combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the Pairwise Gaussian Loss (PGL) [56], to take advantage of the PGL properties, which were mentioned in Section 3.1.2.

3.3.5 MV-Res-SincNet-FC

The proposed MV-Res-SincNet-FC combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the MV-Softmax loss [42], to exploit the merits of the MV-Softmax loss, which were mentioned in Section 3.1.3. The MV-Softmax loss [42] has two versions, the MV-AM-Softmax loss, which uses the same margin of the AM-Softmax loss [21] with the MV-Softmax loss, and MV-Arc-softmax loss, which uses the same margin of the ArcFace loss [22] with the MV-Softmax loss. Therefore, two models are proposed, the MV-AM-Res-SincNet-FC and the MV-Arc-Res-SincNet-FC, which combines the Res-SincNet-FC architecture [2] (depicted in Fig. 3.2) with the MV-AM-Softmax loss and the MV-Arc-Softmax loss, respectively.

3.3.6 D-Res-SincNet-FC

The proposed D-Res-SincNet-FC combines the Res-SincNet-FC architecture (depicted in Fig. 3.3) with the D-Softmax loss [57], to utilize the attributes of the D-Softmax loss, which were mentioned in Section 3.1.4.

Chapter 4

Experimentation and Results

In this chapter, I will go through the implementation details of the experiments conducted in this thesis, and the results achieved by the different models. In the beginning of this chapter, I will discuss the experimental setup, where I will go over the used datasets, the evaluation metrics that will be reported, and implementation details regarding the speaker recognition task and the speaker diarization task.

After that, I will go through the results attained by the models proposed in Chapter 3 and compare their performance to the existing models. Firstly, I will go over the models that are based on the SincNet architecture [1] followed by the models that are based on the Res-SincNet architecture [2] followed by the models based on the proposed Res-SincNet-FC architecture.

Then, I will thoroughly analyze and visualize the diarization results of the model that achieves the best diarization performance. Finally, I will summarize this chapter.

4.1 Experimental Setup

In this section, I will discuss major experimental details that were used across the experiments in this thesis. Firstly, I will give an overview of the used datasets. Secondly, I will go through the evaluation metrics used to measure and compare the performance of the different models. Thirdly, I will discuss the speaker recognition setup and the optimization settings used. After that, I will explain the speaker diarization setup and the d-vectors extraction. Finally, I will go through some details regarding the code implementation in this thesis.

4.1.1 Datasets

Three datasets were considered in this thesis. The TIMIT [71] and the Librispeech [72] datasets were used for the speaker recognition task, while the AMI dataset [73] was used for the speaker diarization task.

TIMIT

The TIMIT dataset [71] has phonetically rich audio samples from 630 American English speakers of eight main dialects sampled at 16 kHz. The TIMIT dataset was originally designed for developing and evaluating automatic speech recognition systems. However, in the SincNet paper [1], the authors used it for the speaker recognition task. This thesis follows [1] where they used the training chunk from the TIMIT dataset, which contains 462 speakers. Moreover, to facilitate text-independent speaker recognition, all sentences with the same text across the speakers were removed. Each speaker has eight sentences five of which were used for training and three for testing. Similar to [1], each audio file was preprocessed by first removing the silent (non-speech) parts at the start and end of each sentence. Then, the amplitude of each audio file was normalized.

Librispeech

The Librispeech [72] dataset has 1000 hours of English speech, which was obtained from audiobooks, spoken by 2484 speakers sampled at 16 kHz. The Librispeech dataset was originally designed for the speech recognition task. However, the authors of [1] used the Librispeech dataset for speaker recognition, where they created the training and testing sets by randomly selecting files such that each speaker has 12-15 seconds of audio for training, and 2-6 seconds long testing sentences. Moreover, they partitioned sentences with more than 125 ms intermediate silences into several chunks. The same selection as in [1] was used in this thesis. The Librispeech dataset was preprocessed similar to the TIMIT dataset, where the silent sections at the start and end of the sentences were removed, and the amplitude normalized.

AMI

The Augmented Multi-party Interaction (AMI) [73] is a multi-modal dataset that has 100 hours of recorded meetings in the English language. This thesis uses five meetings from the AMI dataset to evaluate the diarization performance of the proposed models, where the meetings are headset mix audio files. In each meeting there is a team of four speakers with different roles discussing a design project. The chosen meetings, which are the one used in [8], and their duration can be seen in Table 4.1.

Table 4.1: The chosen AMI meetings and their duration

Meeting ID	Meeting Duration (minutes:seconds)
IS1000a	26:22
IS1001a	15:09
IS1003b	27:26
IS1003d	35:00
IS1008d	24:40

4.1.2 Evaluation Metrics

Two tasks were considered in this thesis speaker recognition and speaker diarization. In order to evaluate the different models properly, the following evaluation metrics are employed. The Frame Error Rate (FER) and the Sentence Error Rate (SER) are used to measure the speaker recognition performance, while the diarization error rate (DER) is used to measure the speaker diarization performance.

Frame Error Rate (FER)

The Frame Error Rate (FER) measures the ability of a model to correctly classify a given frame of fixed length (similar to [1], a frame of size 200 ms was used in this thesis). It is worth noting that all of the SincNet based models [1, 9, 10, 2], and the proposed models in this thesis are designed to optimize the FER.

Sentence Error Rate (SER)

The Sentence Error Rate (SER) measures the ability of a model to correctly classify a sentence (utterance) of variable length that typically lasts more than a frame duration, where the frame duration is 200 ms in this thesis. The predicted speaker for a given sentence is found using the same method used by the original SincNet paper [1], and it is as follows. First, a sliding window (frame) of length 200 ms and a shift of 50 ms is applied to a given sentence. Then, each frame will be passed through the trained model, and the predictions across the different speakers are found for each frame. Finally, the predictions are averaged across all the frames of the given sentence, and the sentence will be assigned to the speaker with the highest average prediction value.

Diarization Error Rate (DER)

Diarization Error Rate (DER) is widely used for scoring diarization systems. From [74], the DER is defined as:

$$DER = MS + FA + SE \tag{4.1}$$

where MS is the missed speech, which measures the percentage of speech intervals that exist in the ground-truth file, but does not exist in the diarization output. FA is the false alarm, which measures the percentage of speech intervals that exist in the diarization output, but does not exist in the ground-truth file. SE is the speaker error, which measures the percentage of speech segments with wrongly assigned speakers. The miss speech and the false alarm errors are segmentation errors that are mostly caused by the front-end modules of the diarization system, such as the speech activity detection. Since the main concern of this thesis is speaker modeling (speaker embeddings extraction) and following [8], the ground-truth speaker segments were used. Thus, the DER in this thesis will be equal to the speaker error. Moreover, the DER is usually measure with a 250 ms tolerance collar at the beginning and end of each segment [74]. However, in the thesis, similar to [8] the DER was computed without tolerance collars. In addition, the overlapped segments were included in the calculation of the DER for all the experiments performed in this thesis.

4.1.3 Speaker Recognition Setup and Optimization Settings

First of all, it is important to note that the same settings were used for all the compared models in this thesis (i.e. the proposed models and the existing models) in order to facilitate a fair comparison between the different models. Similar to the SincNet paper [1] and all the other paper based on it, such as [9], frames of length 200 ms are used as the input to all the models discussed in this thesis. Then, the input frames will be passed through the network, where the output will be the predicted speaker ID that corresponds to the input frame. It can be seen that the models are being trained to minimize the classification error rate of the input frames (i.e. FER). Following [1], the mel-scale cutoff frequencies are used to initialize the learnable parameters of the sinc-layer, and the famous Glorot (also called Xavier) initialization method [75] is used to initialize the remaining network learnable parameters.

Adam optimizer [76] with a learning rate of 0.001 was used for training all the models. Moreover, similar to [2], early stopping and learning rate scheduler were used to improve the training optimization for all the models. When using the TIMIT dataset, a batch size of 154 was used, and model evaluation was performed every 25 epochs. On the other hand, when the Librispeech dataset was used, a batch size of 250 was used, and model evaluation was performed every five epochs. The batch size for the different datasets was chosen such that the batches have equal (or almost equal) size, since it was found that having a batch with a small number of samples compared to the other batches hindered the performance. To facilitate the reproducibility of the results in this thesis, the random seed was set to 42.

4.1.4 Speaker Diarization Setup and d-vectors Extraction

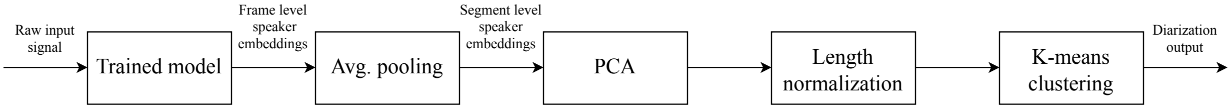

Figure 4.1: The speaker diarization pipeline

The speaker diarization pipeline can be seen in Fig. 4.1. Similar to [8], the different SincNet models are first trained to perform speaker recognition using out-domain data, such as the TIMIT and the Librispeech datasets, then the trained SincNet models are used for speaker embeddings (d-vectors) extraction in speaker diarization system from in-domain data, which is the AMI dataset [73] in this thesis.

In [8], the authors extracted the d-vectors from three different stages: the activations of the last convolutional layer, the activations of the hidden layer, the activations of the classification layer, where they found that extracting the activations from the last hidden layer gave the best performance. Moreover, they tried average pooling and max pooling the frame-level d-vectors, where they found that average pooling yielded better results. Thus, in this thesis, the d-vectors were extracted from the activations of the last hidden layers of the trained models. Additionally, the frame-level d-vectors were average pooled to get the segment-level d-vectors.

Since the goal of this thesis is speaker modeling, and similar to [8], the ground-truth speaker segmentation was used to avoid segmentation errors. Thus, the diarization pipeline is as follows, given a speaker segment, a sliding window (frame) of size 200 ms and a shift of 50 ms, was applied on that segment. Then, each frame was passed through the trained SincNet model, and the frame-level d-vector was extracted from the activations of the last hidden layer. After that, the frame-level d-vectors for that segment were average pooled to get the segment-level d-vector.

Inspired by the d-vector extraction used in [1], and in order to ensure that the extracted segment-level d-vectors are robust, the energy of the frames is taken into consideration before calculating the segment-level d-vectors. The energy of a given frame is found by squaring the magnitude of samples in that frame, then summing them. The frame-level d-vectors extracted from frames that have energy that is less than one-tenth of the average energy of the segment are discarded. In other words, only d-vectors extracted from frames with adequate energy are used to calculate the segment-level d-vectors. Therefore, to make sure that enough frame-level d-vectors (at least 10) are used to get the segment-level d-vector, segments that have a length shorter than two seconds were padded by repeating the segments until two seconds are reached.

Similar to [8], the DER is reported with and without the application of principle component analysis (PCA) [77] on the extracted d-vectors. When PCA is applied, the dimensions of the d-vectors are reduced to 50 dimensions. This thesis uses k-means clustering, with K-means++ initialization [78], to cluster the extracted d-vectors. Finally, similar to [8], length normalization is applied on the embedding vectors before the clustering stage, where each embedding vector was divided by its magnitude before performing k-means clustering.

4.1.5 Code Implementation

All of the codding in this thesis was done using the python programming language. The PyTorch python package [79], which facilitates tensor computation with GPU support, was used extensively in this thesis, such as building and training the models discussed in this thesis. Furthermore, some tools from the scikit-learn library [80] were used, such as the dimensionality reduction using PCA. Moreover, the PyAnnote library [81] was used to calculate the DER performance, and visualize the diarization output. In addition, the authors of [82] provided RTTM (Rich Transcription Time Marked) files corresponding to the AMI dataset [73], which contains the speech intervals of the different speakers for a given meeting in the AMI dataset. In this thesis, the RTTM files provided by [82] were used to extract the ground-truth speaker segmentation from the AMI dataset audio files.

4.2 Results of the SincNet Based Architectures

In this section, I will go through the results of the proposed models based on the SincNet architecture [1] (seen in Fig 3.1), which are the AdaCos-SincNet, PGL-SincNet, MV-SincNet, and D-SincNet. Firstly, I will go over different data augmentation methods that were implemented with the SincNet model. Secondly, I will go through some exploratory studies over the SincNet model, where I will try changing several features of the SincNet model, such as the input window size, and the window shift. Then, I will go through each of the proposed models separately, where I will study the effect of the hyperparameters on the different performance measures. After that, I will compare the performance of the proposed models using the optimum hyperparameters with the existing SincNet based models. Next, I will discuss the performance of the models based on the SincNet architecture when using a relatively small sinc filter size, where a sinc filter size that is equal to 31 samples (1.94 seconds) will be used. Finally, the FER, SER, and DER will be reported using the Librispeech dataset for the models with the best DER performance. The d-vectors will be extracted from the last hidden layer in the SincNet model, which has 2048 neurons. Thus, the d-vectors extracted using the SincNet based models will have 2048 dimensions. These d-vectors will be first used to calculate the DER without PCA. Then, PCA will be applied to the extracted d-vectors to reduce their dimensions to 50, which will be used to calculate the DER with PCA as will be reported in the results tables of this section.

4.2.1 Using Data Augmentation

This section aims to improve the speaker recognition performance of the SincNet model [1] by experimenting with different data augmentation methods. The dataset used in this section is the TIMIT dataset [71] and the evaluation metric considered is the FER. The original authors of SincNet [1] used two augmentation methods, which are:

1. Random framing: Taking a random frame from each sentence
2. Random weighting: Multiplying input frames with a random factor that is between 0.8 and 1.2

Both of these augmentation methods improved the FER performance of the SincNet model. Therefore, using the following standard signal augmentation methods [83] is worth investigating:

1. Additive Gaussian noise: Adding Gaussian noise with 0.004 standard deviation to input frames.
2. Time shifting: Randomly shifting (left or right) input frames in time.
3. Time stretching: Randomly stretching input frames in time (the speed of the audio) with a factor that is between 0.8 and 1.2.
4. Pitch shifting: Randomly shifting (rising or lowering) the pitch of input frames by a semitone that is between -5 and 5 steps.

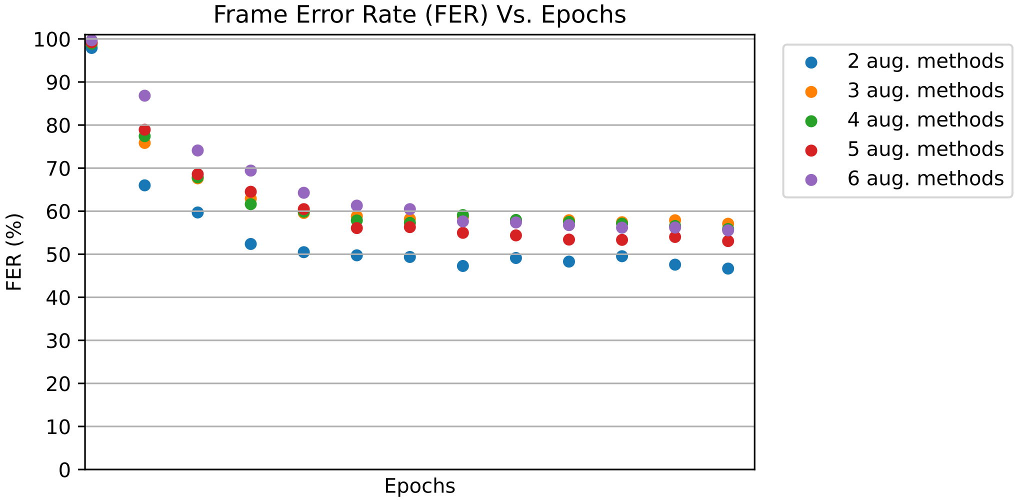

Figure 4.2: The Frame Error Rate (FER) of the different augmentation methods

The following variations of data augmentation were used with the SincNet model:

1. Using two data augmentation methods (random framing and random weighting), which are the two methods used by the original SincNet model [1]. The average epoch time was around 0.42 minutes.
2. Using three data augmentation methods (the methods above plus additive Gaussian noise). The average epoch time was around 0.6 minutes.
3. Using four data augmentation methods (the methods above plus time-shifting). The average epoch time was around 0.7 minutes.

4. Using five data augmentation methods (the methods above plus time stretching). The average epoch time was around 5 minutes.

5. Using six data augmentation methods (the methods above plus pitch stretching). The average epoch time was around 13.6 minutes.

Fig. 4.2 shows the FER achieved by the different data augmentation methods, where it can be seen that the use of the two augmentation methods (random framing and random weighting), which are the ones used in [1], gave the best FER, while also having the least average epoch time. On the other hand, using the additional augmentation methods caused the FER performance to degrade, and they are time-consuming to perform, especially time stretching and pitch stretching, where both have a significantly large average epoch time. Thus, it can be concluded that using further data augmentation methods is bad for the performance of the SincNet model, and they are computationally expensive. The bad performance achieved by the additional augmentation methods could be due to the failure of the model to learn underlying features from the augmented data (i.e. the SincNet model is overfitting the augmented data). Another reason could be the use of several augmentation methods at the same time, which could corrupt the original data, or due to the proposed augmentation methods not being suitable for the speaker recognition task. Thus, the same data augmentation methods in [1] will be used throughout all the experiments in this thesis.

4.2.2 Exploratory Experiments

In this section, I will experiment with different input window shifts, and different input window sizes for the SincNet architecture [1], where I will study their effect on speaker recognition and speaker diarization.

Using different input window shifts

The input window shift is only relevant for the evaluation (testing) phase, since, for the training phase, data augmentation is used, as mentioned in the previous section, where random frames with a window size of 200 ms are used. However, when the SER and the DER are calculated, a sliding window of a specific size and step size (shift) is used to find the sentence-level classification, and the segment-level d-vectors respectively. In the SincNet paper [1], the authors used a window shift size equal to 10ms, and in the other related SincNet papers, such as [9, 2], the authors used the same input shift value. Therefore, in this section, I will experiment with different shift sizes, and analyse the effect of the window step size (shift) on the SER and DER. Table 4.2, shows the speaker recognition, and speaker diarization performances of the SincNet model [1] using different input window shifts. The results in Table 4.2 are on the TIMIT dataset [71].

Table 4.2: The results of the SincNet model using different input shifts (all values are in %)

Input Shift (ms)	FER	SER	DER (w/o PCA)	DER (w/o PCA)
10 [1]	45.64	0.79	30.51	28.77
25	45.73	0.79	30.47	28.70
50	45.67	0.87	30.22	28.88
75	46.07	0.79	-	-

From Table 4.2, when a window shift size of 75 ms was used, causing fewer frames per segment because of the large window shift, the DER could not be calculated due to the lack of enough frame-level d-vectors to calculate the segment-level d-vectors for some of the segments. Moreover, for the 10 ms, 25 ms, and 50 ms cases, it can be seen that the performance is almost the same across all the performance measures, except for the SER, where there is a small difference. Thus, an input window shift size equal to 50 ms will be used, since it gives almost the same performance as smaller shifts, while also greatly reducing the required computational cost.

Using different input window sizes

The input window size is relevant to both the training and evaluation stages. In the SincNet paper [1], and the other related SincNet papers, such as [9, 2], a window size of 200 ms was used. In this section, I run the SincNet model [1] with different input window sizes, and analyse the effect on the speaker recognition and speaker diarization performance measures. Table 4.3, shows the FER, SER, and DER performance measures of the SincNet model [1] with different input window sizes, where the results are reported on the TIMIT dataset [71].

From Table 4.3, it can be seen that an input window size of 400 ms is a better FER compared to the other window sizes, but it gave a significantly worse SER and DER performance. However, when the input window size was 100 ms, the FER performance was significantly worse compared to the other window sizes, but it gave a relatively good performance on the other performance measures. On the other hand, an input window size equal

Table 4.3: The results of the SincNet model using different input window sizes (all values are in %)

Input size (ms)	FER	SER	DER (w/o PCA)	DER (w/o PCA)
100	55.05	0.72	30.08	29.67
200 [1]	45.67	0.87	30.22	28.88
400	41.4	2.89	33.19	33.48

to 200 ms gave the most balanced performance across all the performance measures, where it gave a relatively good performance on the speaker recognition and speaker diarization tasks. Thus, it can be seen that using a relatively small or relatively big window size for the SincNet model [1] can negatively affect different performance measures. Therefore, as in [1], an input window size of 200 ms will be used, which gives a balanced and good performance on all the performance measures, throughout all of the following experiments in this thesis.

Experiments on the Librispeech dataset

In [1], the authors used a sampling frequency (f_s) of 8 kHz, and an input window size of 375 ms when they used the Librispeech dataset for speaker recognition. However, in [11], the authors used a sampling frequency (f_s) of 16 kHz, and a window size of 200 ms, when they used the Librispeech dataset for speaker diarization. In this section, I will compare the speaker recognition and speaker diarization performance of the SincNet model with different combinations of sampling frequencies, and input window sizes, when the Librispeech dataset is used.

From Table 4.4, it can be seen that for a fixed input window size, using a larger sampling frequency ($f_s = 16$ kHz) mostly results in a significantly improved performance on both the speaker recognition and the speaker diarization tasks, since using $f_s = 16$ kHz, results in higher quality audio, which will help the network learn more robust features. Moreover, it can be seen that for $f_s = 16$ kHz, using an input size of 200 ms gives better SER and DER performance. Thus, $f_s = 16$ kHz and input window size equal to 200 ms will be used for all the experiments that involve the Librispeech dataset. Another reason for choosing f_s = 16 kHz and window size = 200 ms, is for the sake of consistency, since both the TIMIT dataset [71] and the AMI dataset [73] have $f_s = 16$ kHz, and a window size of 200 ms is used for the experiments that involve the TIMIT dataset.

Table 4.4: The results of the SincNet model on the Librispeech using different sampling frequencies and different input sizes (all values are in %)

f_s (kHz)	Input size (ms)	FER	SER	DER (w/o PCA)	DER (w/o PCA)
8	200	37.88	0.67	39.31	28.93
16	200	29.34	0.50	30.00	27.21
8	375	36.00	1.06	35.44	26.46
16	375	28.04	0.81	31.80	28.72

4.2.3 AdaCos-SincNet

As discussed in Section 3.1.1, the AdaCos-SincNet model combines the AdaCos loss [55], with the SincNet architecture [1]. The key advantage of the AdaCos loss [55] is the ability to achieve high discriminative performance, without the need for hyperparameter tuning. The AdaCos loss has an s parameter that can be either fixed or dynamic. Table 4.5 shows the results of the AdaCos-SincNet model on the TIMIT dataset [71] using fixed s parameter and dynamic s parameter.

Table 4.5: The results of the AdaCos-SincNet model using different parameters (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Fixed s	38.17	0.72	30.65	33.06
Dynamic s	36.58	0.87	31.26	31.56

From Table 4.5, it can be seen that using the AdaCos-SincNet with dynamic s parameter gave a better FER compared to using a fixed s parameter, since the FER value is being optimized by the network, and using a dynamic s parameter helps in achieving a more optimal value compared to fixed s. Also, it can be observed that the SER and the FER are not correlated, where the AdaCos-SincNet gave a better SER, even though it has a higher FER. With regard to the diarization performance, it can be seen that AdaCos-SincNet with fixed s gave a better performance when PCA was not used and applying PCA degraded its DER performance. However, AdaCos-SincNet with a dynamic s gave a very close DER performance with and without PCA.

4.2.4 PGL-SincNet

As discussed in Section 3.1.2, the PGL-SincNet model combines the Pairwise Gaussian Loss (PGL) [56], with the SincNet architecture [1]. The key advantage of PGL is the fact that it combines the softmax loss with the contrastive loss, which is a metric-based loss. PGL has a single hyperparameter that is β, and Table 4.6 shows the results of the PGL-SincNet model on the TIMIT dataset [71] using three different β values.

Table 4.6: The results of the PGL-SincNet model using different parameters (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$\beta = 0.003$	46.70	1.08	29.63	25.06
$\beta = 0.005$	45.40	0.94	27.69	24.69
$\beta = 0.009$	45.04	1.01	26.33	24.35

From Table 4.6, it can be seen that using $\beta = 0.009$ in the PGL-SincNet model gave the best performance with the respect to the FER and DER for both cases, with and without PCA. This shows that increasing the value of β facilitates the learning of discriminative embeddings, which is seen in the improved DER performance. It can be observed that the SER is not correlated with the FER, where the PGL-SincNet model with $\beta = 0.005$, gave the best SER performance. Also, it can be seen that the DER performance got improved when using PCA for all the cases.

4.2.5 MV-SincNet

As discussed in Section 3.1.3, since there are two versions of the MV-Softmax loss, the MV-AM-Softmax and the MV-Arc-Softmax, two models that utilize the MV-Softmax loss were proposed. The first is the MV-AM-SincNet model, which combines the SincNet architecture [1] with the MV-AM-Softmax loss [42]. The second one is the MV-Arc-SincNet model, which combines the SincNet architecture [1] with the MV-Arc-Softmax loss [42]. The key advantage of the MV-AM-Softmax and the MV-Arc-Softmax losses is the fact that it combines feature mining (i.e. learning more from hard samples) with the feature margin methods, which are the AM-Softmax [21] and the ArcFace [22] methods, respectively.

The MV-Softmax loss [42] mainly has three hyperparameters, which are the value of the margin parameter (m), the value of the t parameter, and if the weighting function of the miss classified samples is fixed or adaptive. However, the authors of [42] found that using an adaptive weighting function gives better performance compared to a fixed one; hence an adaptive weighting function will be used for all the models that use the MV-Softmax loss. Thus, in this section, I will discuss the results of the MV-AM-SincNet model, then the MV-Arc-SincNet model on the TIMIT dataset [71] using different variations of the m and t hyperparameters. It is worth noting that similar to the MV-Softmax paper [42], the scale parameter is set to 32 throughout the different experimentation. Table 4.7 shows the results of the MV-AM-SincNet model with different hyperparameters.

Table 4.7: The results of the MV-AM-SincNet model using different parameters (all values are in %)

Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.35, t = 0.2$	36.22	0.58	23.83	21.87
$m = 0.35, t = 0.02$	38.94	0.79	29.53	24.86
$m = 0.5, t = 0.2$	35.62	0.65	32.38	26.90

From Table 4.7, it can be seen that the MV-AM-SincNet model with $m = 0.5$ and $t = 0.2$ gave the best FER, but it gave a relatively poor DER performance. Also, it can be observed that the MV-AM-SincNet model with $m = 0.35$ and $t = 0.2$ gave the best DER performance with and without PCA. Additionally, for all cases, it can be seen that applying PCA improved the DER performance. Table 4.8 shows the results of the MV-Arc-SincNet model with different hyperparameters.

Table 4.8: The results of the MV-Arc-SincNet model using different parameters (all values are in %)

Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.5, t = 0.2$	39.26	0.51	23.48	24.80
$m = 0.5, t = 0.02$	38.57	0.72	31.91	25.47
$m = 0.5, t = 0.3$	36.50	0.58	31.61	27.23

From Table 4.8, it can be seen that the MV-Arc-SincNet model with $m = 0.5$ and $t = 0.3$ gave FER performance. On the other hand, it can be observed that the MV-Arc-SincNet model with $m = 0.5$ and $t = 0.2$ gave the best performance on the SER and

DER for both cases, with and without PCA. Also, almost in all cases, it can be seen that applying PCA improved the DER performance.

4.2.6 D-SincNet

As discussed in Section 3.1.4, the D-SincNet model combines the D-Softmax loss [57] with the SincNet architecture [1]. The key advantage of the D-Softmax loss is the fact that it disentangles the conventional softmax loss into inter-class objective and intra-class objective, where it optimizes both objectives independently which improves the ability to extract discriminative features. The D-Softmax mainly has a single hyperparameter, which is the d parameter. It is worth noting that similar to the D-Softmax paper [57], the scaling parameter (s) is set to 32, throughout all the experiments. Table 4.9 shows the results of the D-SincNet model with different d values.

Table 4.9: The results of the D-SincNet model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$d = 0.3$	35.47	0.72	30.72	27.25
$d = 0.5$	36.08	0.36	34.65	25.69
$d = 0.9$	35.76	0.65	29.14	24.36

From Table 4.9, it can be seen that the different D-SincNet models gave almost similar FER performance. However, the D-SincNet model with $d = 0.5$ gave significantly good SER performance. Whereas, the D-SincNet model with $d = 0.9$ gave the best DER performance for both cases, with and without PCA. This indicates that a large d value can facilitate better speaker embeddings extraction, since it gave a better DER performance compared to using a smaller d value.

4.2.7 Comparing the Models with the Best DER

After experimenting on the proposed models based on the SincNet architecture with different hyperparameters, the hyperparameters that gave the best overall DER performance were selected for each of the proposed models. Then, the selected models and the existing models were compared with respect to the FER, SER, and DER with and without PCA. Firstly, the existing SincNet models that are concerned with the speaker recognition task, which are the SincNet [1], AM-SincNet [9], and AF-SincNet [10], were trained using the same optimization settings discussed in Section 4.1.3. This will ensure that the comparison is fair between all the models, where the only thing changing is the used loss with the SincNet architecture while everything else is the same across all the models. The SincNet model [1] uses the conventional softmax loss which does not have a hyperparameter to tune. However, for the AM-SincNet model [9] which uses the AM-Softmax loss [21], and for the AF-SincNet model [10] which used the ArcFace loss [22], the margin parameter m is set to 0.5 and the scale parameter s to 30 for both models, which are the values chosen in the respective papers of both models.

Secondly, there are two existing SincNet based models that are concerned with the speaker diarization task. In [8] the authors used the SincNet model [1] for speaker embeddings extraction in a speaker diarization system, while in the [11] authors used the AM-SincNet model [9] for speaker embeddings extraction in a speaker diarization system. For both of the previous cases, the same procedure discussed in Section 4.1.4 is used to ensure a fair comparison across all the models. It is worth noting that even though the AF-SincNet [10] was used for speaker recognition, no one proposed using it for speaker diarization. Thus, this thesis proposes using the AF-SincNet model [10] for speaker embeddings extraction in a speaker diarization system. Table 4.10 shows the results of the proposed models compared to the existing models, with their respective chosen hyperparameters. The results in Table 4.10 are reported on the TIMIT dataset [71].

Table 4.10: Comparing the results of the proposed and existing models that are based on the SincNet architecture (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
SincNet [1], [8]	-	45.67	0.87	30.22	28.88
AM-SincNet [9], [11]	$m = 0.5$	38.69	*0.65*	29.71	26.24
AF-SincNet [10]	$m = 0.5$	37.57	0.72	*25.42*	25.52
AdaCos-SincNet	Dynamic s	*36.58*	0.87	31.26	31.56
PGL-SincNet	$\beta = 0.009$	45.04	1.01	26.33	25.35
MV-AM-SincNet	$m = 0.35, t = 0.2$	36.22	0.58	23.83	**21.87**
MV-Arc-SincNet	$m = 0.5, t = 0.2$	39.26	**0.51**	**23.48**	*24.80*
D-SincNet	$d = 0.9$	**35.76**	*0.65*	29.14	24.36

For each of the evaluation metrics in Table 4.10, the best values are in bold, the second-best values are underlined, while the third-best values are in italic. Referring to Table 4.10, it can be seen that the proposed models achieved the top three performances across the different evaluation metrics. The MV-AM-SincNet achieved the best DER performance when PCA was used, and it achieved the second-best performance on FER, SER, and DER when PCA was not used. In addition, the MV-Arc-SincNet model achieved the best SER and DER when PCA was not used, and the third-best performance on the DER when PCA was used. This indicates that the MV-Softmax loss [42] is the best loss for extracting discriminative embeddings, since the two models based on it, which are the MV-AM-SincNet and the MV-Arc-SincNet gave significant DER improvement compared to the other methods, while also giving the best SER performance for the speaker recognition task. Furthermore, the D-SincNet model gave the third-best SER performance which is equal to the SER achieved by the AM-SincNet model [9]; however, the D-SincNet achieved the best FER and the second-best DER when PCA was used.

In general, from Table 4.10, it can be seen a correlation between the SER performance and the DER performance, where the models that achieved high SER performance also achieved high DER performance. For example, the MV-Arc-SincNet model achieved the best SER performance, and it also achieved the best DER performance for the case without PCA, while also achieving a top three DER performance when PCA was used. Moreover, referring to Table 4.10, it can be seen that for most cases using PCA improved the DER performance.

4.2.8 Results Using Sinc Filter Size of 31

In the Res-SincNet paper [2], the authors performed a thorough analysis of the sinc filter's window size and stride (step size), where they evaluated the Res-SincNet model using a wide range of sinc filter's window sizes and strides. They found that using a stride of one gives the best performance, which is the same stride used in the original SincNet paper [1], and it is used for all the models in this thesis. However, the author's in [2], also found that the sinc filter window size slightly affects the FER performance of the Res-SincNet model. Therefore, they proposed reducing the sinc filter window size from 251 samples (15.69 ms) (which is the value used in the original SincNet paper [1]) to 31 samples (1.94 ms), which will reduce the computational cost without hindering the FER performance. Thus, inspired by the previously mentioned findings from [2], all the models based on the SincNet architecture were evaluated using a sinc filter size of 31. For the proposed models, the chosen hyperparameters are the ones that give the best FER and the ones that give the best the SER. Then, they were compared with the SincNet model [1], the AM-SincNet model [9] with $m = 0.5$, and the AF-SincNet model [10] with $m = 0.5$, which are the optimum m values used in their respective papers. Table 4.11 shows the results of all the SincNet based models with sinc filter window size equal to 31 samples.

Table 4.11: The results of the models based on the SincNet architecture with sinc filter = 31 (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
SincNet [1]	-	41.77	0.87	34.86	35.09
AM-SincNet [9]	$m = 0.5$	32.5	0.58	32.68	33.26
AF-SincNet [10]	$m = 0.5$	32.89	0.65	33.91	36.70
AdaCos-SincNet	Fixed s	31.79	0.72	36.35	32.89
AdaCos-SincNet	Dynamic s	31.93	0.65	32.05	35.88
PGL-SincNet	$\beta = 0.005$	42.32	0.79	34.19	31.99
PGL-SincNet	$\beta = 0.009$	43.02	0.87	33.15	33.29
MV-AM-SincNet	$m = 0.35, t = 0.2$	35.38	0.65	30.73	30.03
MV-AM-SincNet	$m = 0.5, t = 0.2$	31.96	0.65	31.74	32.00
MV-Arc-SincNet	$m = 0.5, t = 0.2$	32.03	0.58	33.01	34.65
MV-Arc-SincNet	$m = 0.5, t = 0.3$	32.93	0.65	35.09	35.43
D-SincNet	$d = 0.3$	31.81	0.43	33.05	35.01
D-SincNet	$d = 0.5$	32.33	0.36	32.28	31.70

From Table 4.11, it can be seen that for all the models the FER performance is better compared to the results in Table 4.10, where a sinc filter window size of 251 samples was used. This indicates that reducing the sinc filter size from 251 to 31 improves the FER performance. However, it can be observed that the DER performance is significantly worse compared to the results in Table 4.10, which indicates that reducing the sinc filter size from 251 to 31 impacted the DER performance negatively. On the other hand, the SER performance somewhat stayed the same compared to using a sinc filter size equal to 251 samples.

Referring to Table 4.11, it can be seen that the AdaCos-SincNet model gave the best FER performance, but it is very close to the FER performance achieved by most of the other models, where it can be observed that most of the models achieved a FER that is almost equal to 32%. This similar performance can be due to the network architecture, which could be acting as a bottleneck by limiting the classification performance of the different losses. Furthermore, it can be seen from Table 4.11 that the D-SincNet models gave an exceptionally good SER performance, where both D-SincNet models in Table 4.11 achieved the best two SER performances. In addition, it can be seen that the MV-AM-SincNet model with $m = 0.35$ and $t = 0.2$ gave the best DER performance, which is also the same model that achieved the best DER performance in Table 4.10. This shows the robustness of the MV-AM-SincNet model, where it still gave the best DER compared to the other models, even after reducing the sinc filter size to 31.

To sum up, using a small sinc filter size, reduced the computational cost significantly for both the speaker recognition task and the speaker diarization task, where it gave an improvement over the FER performance while degrading the DER performance. Moreover, the SincNet network architecture could be limiting the classification performance of the improved loss functions.

4.2.9 Results on the Librispeech Dataset

In this section, the Librispeech dataset [72], which is a relatively large dataset, is used to train and evaluate the models discussed in Section 4.2.7, where the hyperparameters that gave the best DER performance were chosen. This will help us study the effect of using a large dataset on the different performance measures across the different models. Table 4.12 shows the results of the different SincNet based models with the same settings as in Table 4.10, where the Librispeech dataset [72] is used.

Table 4.12: Comparing the results of the proposed and existing models that are based on the SincNet architecture using the Librispeech dataset (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
SincNet [1], [8]	-	29.34	0.50	30.00	27.21
AM-SincNet [9], [11]	$m = 0.5$	**17.60**	0.39	*23.65*	*23.88*
AF-SincNet [10]	$m = 0.5$	*18.28*	0.44	23.63	*23.88*
AdaCos-SincNet	Dynamic s	19.66	0.48	**23.29**	27.45
PGL-SincNet	$\beta = 0.009$	32.00	0.54	30.18	26.44
MV-AM-SincNet	$m = 0.35, t = 0.2$	19.32	**0.35**	24.36	**19.52**
MV-Arc-SincNet	$m = 0.5, t = 0.2$	18.17	0.44	23.85	23.32
D-SincNet	$d = 0.9$	21.67	*0.43*	26.94	26.12

For each of the evaluation metrics in Table 4.12, the best values are in bold, the second-best values are underlined, while the third-best values are in italic. From Table 4.12, it can be seen that there is a significant improvement of the FER and SER for all the models, compared to the same models in Table 4.10. This shows that using a large dataset can greatly improve the speaker recognition performance of the SincNet based models. Moreover, it can be observed that most models in Table 4.12 got an improved DER performance compared to their performance on the TIMIT dataset reported in Table 4.10. This indicates that models trained on large datasets, such as the Librispeech dataset, are better when used for embeddings extraction. This can be explained by the fact that, when a large dataset is used, the SincNet models become more capable of learning more generalized features. Thus, these trained models can extract more discriminative and robust embeddings, when used for speaker modeling, and hence resulting in a better speaker diarization performance.

Referring to Table 4.12, it can be seen that for each performance metric the performance of the top three models is somewhat close, except for DER with PCA, where it can be observed that the MV-AM-SincNet model gave a significantly improved performance. This shows the robustness and superiority of the MV-AM-SincNet on the speaker diarization task, since it was consistently achieving the best DER compared to the other models under different conditions.

4.3 Results of the Res-SincNet Based Architectures

As discussed in Section 3.2, various models based on the Res-SincNet architecture [2] (seen in Fig. 3.2) were proposed, which are:

1. AM-Res-SincNet model, which combines the AM-Softmax loss [21], with the Res-SincNet architecture [2]

2. Arc-Res-SincNet model, which combines the ArcFace loss [22], with the Res-SincNet architecture [2]

3. AdaCos-Res-SincNet model, which combines the AdaCos loss [55], with the Res-SincNet architecture [2]

4. PGL-Res-SincNet model, which combines the Pairwise Gaussian Loss (PGL) [56], with the Res-SincNet architecture [2]

5. MV-AM-Res-SincNet model, which combines the MV-AM-Softmax loss [42], with the Res-SincNet architecture [2]

6. MV-Arc-Res-SincNet model, which combines the MV-Arc-Softmax loss [42], with the Res-SincNet architecture [2]

7. D-Res-SincNet model, which combines the D-Softmax loss [21], with the Res-SincNet architecture [2]

Table 4.13 shows the performance of the Res-SincNet model [2], which uses the conventional softmax loss, compared to the aforementioned proposed models based on the Res-SincNet architecture [2]. Table 4.13 shows the chosen hyperparameters for proposed models, where all the models are trained on the TIMIT dataset [71]. Also, it is worth noting that the Res-SincNet model was used for speaker recognition only, but this thesis proposes using it for speaker diarization. The d-vectors will be extracted from the penultimate layer, which is the global average pooling layer in the Res-SincNet architecture. The global average pooling layer is of size 512, so, the extracted d-vectors will have 512 dimensions. These d-vectors will be first used to calculate the DER without PCA. Then, PCA will be applied on the extracted d-vectors to reduce their dimensions to 50, which will be used to calculate the DER with PCA as will be reported in the results tables of this section.

Table 4.13: The results of the Res-SincNet based models (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet [2]	-	**28.3**	1.15	44.18	46.12
AM-Res-SincNet	$m = 0.35$	30.34	1.08	42.22	43.75
Arc-Res-SincNet	$m = 0.5$	30.22	1.30	49.83	45.86
AdaCos-Res-SincNet	fixed s	35.44	2.96	46.86	46.81
PGL-Res-SincNet	$\beta = 0.005$	28.69	**0.58**	47.00	45.72
MV-AM-Res-SincNet	$m = 0.35, t = 0.02$	30.04	1.08	45.47	43.38
MV-Arc-Res-SincNet	$m = 0.5, t = 0.03$	31.38	1.73	43.37	44.32
D-Res-SincNet	$d = 0.5$	28.66	1.15	43.44	42.90

Firstly, considering the models' performance on the speaker recognition task (i.e. FER and the SER columns) in Table 4.13, where the best values are in bold. It can be seen that most of the proposed models gave a bad performance compared to the Res-SincNet model [2]. This can be explained by the fact that in the original papers of the advanced softmax based losses, such as the ArcFace loss [22], the authors replaced the global average pooling layer, which is the penultimate layer, in the ResNet architecture [23] with a fully connected (FC) layer when they combined it with their losses. Replacing the global average pooling layer with an FC layer, will allow the advanced losses to learn high-level discriminative features in the penultimate layer. Since, the Res-SincNet architecture [2] is inspired by the ResNet architecture [23], the global average pooling layer in the Res-SincNet model should be replaced with an FC layer, which is why the Res-SincNet-FC model (discussed in Section 3.3) was proposed. It is worth noting that the PGL-Res-SincNet model gave an improved SER performance compared to the Res-SincNet model [2], since PGL can be used with the ResNet architecture without the need for replacing the global average pooling layer with an FC layer.

Secondly, considering the models' performance on the speaker diarization task, which is measure using DER. From Table 4.13, it can be seen that all of the models gave a low DER performance, which could be caused by the use of a relatively small sinc filter size (89 samples (5.56 ms)) in the Res-SincNet architecture. However, since the Res-SincNet architecture [2] is not suitable to be combined with the advanced softmax based losses, such as AM-Softmax [21], investigating larger sinc filter sizes will be only performed for the Res-SincNet-FC architecture, which will be discussed in the following section.

4.4 Results of the Res-SincNet-FC Based Architectures

In this section, I will discuss different models based on the proposed Res-SincNet-FC architecture (seen in Fig. 3.3), which is a modified Res-SincNet architecture [2] that is more suitable to be combined with advanced softmax losses. The Res-SincNet-FC architecture has the same architecture as the Res-SincNet architecture [2], but with the global average pooling layer replaced by the following layers: a batch normalization [69], dropout [70], fully connected, batch normalization [69], and leaky ReLU [68] activation. Different variations and combinations of the fully connected layer with other types of layers and activations were investigated, and it was found that the BN-dropout-FC-BN-leaky ReLU layers gives the best performance. The replacement of the global average pooling layer with BN-dropout-FC-BN-leaky ReLU layers is inspired by the architecture used in the ArcFace paper [22], where they replaced the global average pooling layer in the ResNet architecture [23] with BN-dropout-FC-BN layers. The d-vectors will be extracted from the activations of the last hidden layer, which has a size of 512. Thus, the extracted d-vectors will have 512 dimensions. These d-vectors will be first used to calculate the DER without PCA. Then, PCA will be applied on the extracted d-vectors to reduce their dimensions to 50, which will be used to calculate the DER with PCA as will be reported in the results tables of this section.

4.4.1 Res-SincNet-FC

The Res-SincNet-FC model uses the conventional softmax loss similar to the Res-SincNet model [2], which will make sure that the different losses are compared using the same architecture. Thus, the Res-SincNet-FC model can serve as a baseline to compare the effect of using modified loss, when the Res-SincNet-FC architecture is used. Table 4.14 shows the speaker recognition and speaker diarization performance of the Res-SincNet [2] and the Res-SincNet-FC models on the TIMIT dataset [71].

Table 4.14: Comparing the Res-SincNet and the Res-SincNet-FC models (all values are in %)

Model	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet [2]	28.30	1.15	44.18	46.12
Res-SincNet-FC	28.68	1.15	48.33	46.31

From Table 4.14, it can be seen that modifying the architecture of the Res-SincNet [2] model is not enough to enhance its performance, since both models in Table 4.14 has the conventional softmax loss, but the Res-SincNet-FC has a modified architecture, and it did not improve the performance over the Res-SincNet model [2]. Moreover, it can be observed that both models gave a low diarization performance, which could be due to the relatively small size of the sinc filter. Therefore, the size of the sinc filter of both models was increased to 251 samples and analyze the effect on the diarization performance, which can be seen in Table 4.15.

From Table 4.15, it can be seen that increasing the size of the sinc filter size greatly improves the diarization performance, especially for the Res-SincNet-FC model. This indicates the importance of using a large sinc filter size to ensure a high Res-SincNet-FC diarization performance. However, it can be observed that increasing the sinc filter size to 251 samples mildly degrades the speaker recognition performance. Furthermore, it can be

Table 4.15: Comparing the Res-SincNet and the Res-SincNet-FC models with sinc filter size of 251 samples (all values are in %)

Model	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet [2]	32.90	1.59	37.86	40.99
Res-SincNet-FC	32.51	1.52	35.67	34.43

seen that after increasing the since filter size, the Res-SincNet-FC gave a slightly better FER and SER, but it gave significantly better DER, particularly when PCA was applied, compared to the Res-SincNet model [2]. This shows the superiority of the Res-SincNet-FC model to extract better speaker embeddings compared to the Res-SincNet model [2], when the right settings are used (i.e. a suitable sinc filter size). Thus, it is sensible to concentrate only on improving the Res-SincNet-FC model by combining it with different advanced loss functions, which will be discussed in further sections.

4.4.2 AM-Res-SincNet-FC

As discussed in Section 3.3.1, the AM-Res-SincNet-FC model combines the AM-Softmax loss [21] with the Res-SincNet-FC architecture. Table 4.16 shows the speaker recognition and speaker diarization performance of the AM-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameter m values.

Table 4.16: The results of the AM-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.2$	23.67	0.51	48.44	48.17
$m = 0.35$	23.53	0.58	45.31	43.45
$m = 0.5$	24.39	1.01	43.32	41.31

From Table 4.16, it can be seen that using $m = 0.2$ and $m = 0.35$ gave a comparable speaker recognition performance, but using $m = 0.35$ gave a better diarization performance. On the other hand, using $m = 0.5$ gave the best diarization performance.

Table 4.17 shows the speaker recognition and diarization performance of the AM-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.17

are on the two models with the hyperparameter choices that give the best DER performance from Table 4.16, where a sinc filter size of 89 samples was used.

Table 4.17: The results of the AM-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.35$	25.85	1.01	39.59	36.62
$m = 0.5$	28.41	1.66	36.98	34.50

From Table 4.17, it can be seen that using $m = 0.5$ gives the best diarization performance, which is the same hyperparameter setting that gives the best DER when a sinc filter size of 89 samples was used. This shows that using a relatively small m value for the AM-Res-SincNet-FC is bad for the diarization performance.

4.4.3 Arc-Res-SincNet-FC

As discussed in Section 3.3.2, the Arc-Res-SincNet-FC model combines the ArcFace loss [22] with the Res-SincNet-FC architecture. Table 4.18 shows the speaker recognition and speaker diarization performance of the Arc-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameter m values.

Table 4.18: The results of the Arc-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.2$	24.98	0.58	49.15	44.49
$m = 0.35$	24.18	0.72	41.71	42.56
$m = 0.5$	24.62	1.15	43.18	45.09

From Table 4.18, it can be seen that the three m-values gave a similar FER performance, but $m = 0.2$ gave the best SER performance. However, $m = 0.35$ gave the best DER performance.

Table 4.19 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.19

are on the two models with the hyperparameter choices that give the best DER performance from Table 4.18, where a sinc filter size of 89 samples was used.

Table 4.19: The results of the Arc-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.35$	26.06	0.51	40.24	32.99
$m = 0.5$	27.17	1.44	35.16	31.97

From Table 4.19, it can be seen that using $m = 0.5$ gives the best diarization performance. This shows that using a relatively small m value for the Arc-Res-SincNet-FC is bad for the diarization performance.

4.4.4 AdaCos-Res-SincNet-FC

As discussed in Section 3.3.3, the Arc-Res-SincNet-FC model combines the AdaCos loss [55] with the Res-SincNet-FC architecture. Table 4.20 shows the speaker recognition and speaker diarization performance of the AdaCos-Res-SincNet-FC on the TIMIT dataset [71], using fixed s parameter and dynamic s parameter.

Table 4.20: The results of the AdaCos-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
fixed s	25.48	1.37	39.76	35.75
dynamic s	26.34	1.44	41.44	43.16

From Table 4.20, the AdaCos-Res-SincNet-FC model with a fixed s parameter gave a better performance across all measures.

Table 4.21 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.21 are on the two models with the hyperparameter choices that give the best DER performance from Table 4.20, where a sinc filter size of 89 samples was used.

Table 4.21: The results of the AdaCos-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
fixed s	27.83	1.52	36.24	33.18
dynamic s	28.36	1.73	41.04	38.35

From Table 4.21, it can be seen that using fixed s parameter gives the best diarization performance, which also gave the best DER when sinc filter size equal to 89 samples was used. This indicates that using a fixed s parameter for the AdaCos-Res-SincNet-FC is suitable for achieving high DER performance.

4.4.5 PGL-Res-SincNet-FC

As discussed in Section 3.3.4, the PGL-Res-SincNet-FC model combines the Pairwise Gaussian Loss (PGL) [56] with the Res-SincNet-FC architecture. Table 4.22 shows the speaker recognition and speaker diarization performance of the PGL-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameter β values.

Table 4.22: The results of the PGL-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$\beta = 0.003$	30.94	1.59	48.57	44.43
$\beta = 0.005$	28.17	1.37	53.51	46.42
$\beta = 0.009$	28.02	1.23	45.53	43.94

From Table 4.22, it can be seen that the PGL-Res-SincNet-FC with $\beta = 0.009$ performed the best on all the performance measures.

Table 4.23 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.23 are on the two models with the hyperparameter choices that give the best DER performance from Table 4.22, where a sinc filter size of 89 samples was used.

Table 4.23: The results of the PGL-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$\beta = 0.003$	33.65	1.95	34.93	32.20
$\beta = 0.009$	31.67	1.44	43.73	38.10

From Table 4.23, it can be seen that using $\beta = 0.003$ gives the best diarization performance, which shows that using a relatively small β value is good for the diarization performance.

4.4.6 MV-Res-SincNet-FC

As discussed in Section 3.3.5, the MV-Softmax loss [42] has two versions, the MV-AM-Softmax and the MV-Arc-Softmax. Thus, two models based on the MV-Softmax loss [42] are proposed. The first is the MV-AM-Res-SincNet-FC model, which combines the Res-SincNet-FC architecture with the MV-AM-Softmax loss [42]. The second one is the MV-Arc-Res-SincNet-FC model, which combines the Res-SincNet-FC architecture with the MV-Arc-Softmax loss [42].

As mentioned previously, the MV-Softmax loss has three main hyperparameters, which are the value of the margin parameter (m), the value of the t parameter, and if the weighting function of the miss classified samples is fixed or adaptive. In this section, only the m and t parameters will be tuned. In addition, the dynamic weighting function will be used for all the models since, according to its original paper, it gives a better performance compared to using a fixed weighting function. It is worth noting that similar to the MV-Softmax paper [42], the scale parameter is set to 32 throughout the different experimentation.

Table 4.24 shows the speaker recognition and speaker diarization performance of the MV-AM-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameters m and t values.

From Table 4.24, it can be seen that the best performance was achieved using the models that use $m = 0.35$ with $t = 0.02$ and $m = 0.35$ with $t = 0.03$, where they gave a comparable FER and DER performance, but using $m = 0.35$ with $t = 0.03$ gives a better SER performance.

Table 4.24: The results of the MV-AM-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.35, t = 0.02$	23.31	0.94	43.48	45.80
$m = 0.35, t = 0.03$	23.65	0.79	46.76	42.62
$m = 0.5, t = 0.02$	24.22	0.94	51.37	47.59
$m = 0.35, t = 0.2$	24.93	1.01	49.36	47.95

Table 4.25 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.25 are on the two models with the hyperparameter choices that give the best DER performance from Table 4.24, where a sinc filter size of 89 samples was used.

Table 4.25: The results of the MV-AM-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.35, t = 0.02$	26.14	1.08	39.44	40.42
$m = 0.35, t = 0.03$	26.49	0.94	39.68	44.35

From Table 4.25, it can be seen that both models gave a similar DER performance when PCA was not used, but when PCA was used, the DER performance of the model using $m = 0.35$ with $t = 0.02$ was better, which shows that using a relatively small t value for the same m value ($m = 0.35$) is good for improving the diarization performance of the MV-AM-Res-SincNet-FC model.

Table 4.26 shows the speaker recognition and speaker diarization performance of the MV-Arc-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameters m and t values.

From Table 4.26, it can be seen that the best performance was achieved using the models that use $m = 0.5$ with $t = 0.02$ and $m = 0.5$ with $t = 0.03$, where they gave a comparable FER and DER performance, but using $m = 0.5$ with $t = 0.02$ gives a better SER performance.

Table 4.27 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.27

Table 4.26: The results of the MV-Arc-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.5, t = 0.02$	25.15	0.87	46.73	45.47
$m = 0.5, t = 0.03$	24.71	1.15	46.45	43.98
$m = 0.35, t = 0.03$	24.02	0.94	51.25	54.21
$m = 0.5, t = 0.3$	24.47	1.23	48.61	49.93

are on the two models with the hyperparameter choices that give the best DER performance from Table 4.26, where a sinc filter size of 89 samples was used.

Table 4.27: The results of the MV-Arc-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$m = 0.5, t = 0.02$	27.01	1.01	34.44	36.85
$m = 0.5, t = 0.03$	27.50	1.30	37.75	33.62

From Table 4.27, it can be seen that $m = 0.5$ with $t = 0.02$ gave a better DER compared to $m = 0.5$ with $t = 0.03$ when PCA was not used, while $m = 0.5$ with $t = 0.03$ gave a better DER when PCA was used. In general, it can be observed that both models in Table 4.27 gave somewhat a similar overall DER performance, where one gave a good performance when PCA was not used, while the other performed well when PCA was used.

4.4.7 D-Res-SincNet-FC

As discussed in Section 3.3.6, the D-Res-SincNet-FC model combines the D-Softmax [57] with the Res-SincNet-FC architecture. Table 4.28 shows the speaker recognition and speaker diarization performance of the D-Res-SincNet-FC on the TIMIT dataset [71], using different hyperparameter d values, while using s parameter equal to 32, similar to [57].

From Table 4.28, it can be seen that the best speaker recognition performance was achieved using $d = 0.3$ and $d = 0.5$, where they gave a comparable performance. However, the best speaker diarization performance was achieved using $d = 0.3$ and $d = 0.9$, where they also gave a comparable performance.

Table 4.28: The results of the D-Res-SincNet-FC model (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$d = 0.3$	22.79	0.94	43.06	43.72
$d = 0.5$	23.13	0.79	47.61	52.81
$d = 0.9$	26.07	1.30	43.63	42.16

Table 4.29 shows the speaker recognition and diarization performance of the Arc-Res-SincNet-FC model when a sinc filter size of 251 is used. The reported results in Table 4.29 are on the two models with the hyperparameter choices that give the best DER performance from Table 4.28, where a sinc filter size of 89 samples was used.

Table 4.29: The results of the D-Res-SincNet-FC model with sinc filter size equal 251 (all values are in %)

Hyperparameter	FER	SER	DER (w/o PCA)	DER (w/ PCA)
$d = 0.3$	25.44	0.87	43.18	38.74
$d = 0.9$	29.41	1.52	35.31	37.02

From Table 4.27, it can be seen that $d = 0.9$ gave the best diarization performance, which shows that a large d value can facilitate better speaker embeddings extraction, since it gave a better DER performance compared to using a small d value.

4.4.8 Comparing the Models with the Best FER

As seen from the previous sections, using a sinc filter size of 89 samples gives a better speaker recognition performance, especially on the frame-level classification task (measure using the FER), compared to using a size of 251 samples. Thus, this section concentrates on the Res-SincNet-FC based models that gave the best speaker recognition performance, where for each Res-SincNet-FC based model, the chosen hyperparameters are the ones that gave the best FER and SER. It is worth noting that several models have two hyperparameter choices, where one gives better FER and the other gives a better SER, while other models have a hyperparameter setting that performed the best on both FER and SER. Table 4.30 shows the speaker recognition and speaker diarization performance of the different Res-SincNet-FC based models with sinc filter size equal to 89 samples, and the hyperparameters that give the best speaker recognition performance for each model. The results in Table 4.30 are on the TIMIT dataset [71].

Table 4.30: The results of the models based on the Res-SincNet-FC architecture with sinc filter = 89 (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet-FC	-	28.68	1.15	48.33	46.31
AM-Res-SincNet-FC	$m = 0.2$	23.67	**0.51**	48.44	48.17
AM-Res-SincNet-FC	$m = 0.35$	23.53	*0.58*	45.31	43.45
Arc-Res-SincNet-FC	$m = 0.2$	24.98	*0.58*	49.15	44.49
Arc-Res-SincNet-FC	$m = 0.35$	24.18	0.72	*41.71*	*42.56*
AdaCos-Res-SincNet-FC	Fixed s	25.48	1.37	**39.76**	**35.75**
PGL-Res-SincNet-FC	$\beta = 0.009$	28.02	1.23	45.53	43.94
MV-AM-Res-SincNet-FC	$m = 0.35, t = 0.02$	23.31	0.94	43.48	45.80
MV-AM-Res-SincNet-FC	$m = 0.35, t = 0.03$	23.65	0.79	46.76	42.62
MV-Arc-Res-SincNet-FC	$m = 0.5, t = 0.02$	25.15	0.87	46.73	45.47
MV-Arc-Res-SincNet-FC	$m = 0.35, t = 0.03$	24.02	0.94	51.25	54.21
D-Res-SincNet-FC	$d = 0.3$	**22.79**	0.94	43.06	43.72
D-Res-SincNet-FC	$d = 0.5$	*23.13*	0.79	47.61	52.81

In Table 4.30, the best value for each performance measure is in bold, while the second-best is in italic. From Table 4.30, it can be seen that the D-Res-SincNet-FC gives the best FER performance, since it was able to achieve the best and second-best FER values, using different d values. However, the AM-Res-SincNet-FC gave the best SER performance, where it achieved the best SER performance using $m = 0.2$, and it gave the second-best SER performance using $m = 0.35$, along with the Arc-Res-SincNet-FC that has $m = 0.2$. Most of the models in Table 4.30 gave a low DER performance, since they are using a relatively small sinc filter size. However, it can be observed that the AdaCos-Res-SincNet-FC gave a significantly better DER performance compared to the other models in Table 4.30. The Arc-Res-SincNet-FC model with $m = 0.35$ gave the second-best DER performance.

4.4.9 Comparing the Models with the Best DER

As seen from the previous sections, using a sinc filter size of 251 samples gives a better speaker diarization performance compared to using a size of 89 samples, which gives a relatively better speaker recognition performance. In the previous section (Section 4.4.8) the Res-SincNet-FC based models with the best speaker recognition performance were compared, so a sinc filter size of 89 was used. In this section, the performance of the Res-SincNet-FC based models will be compared on the diarization task, where a sinc filter size of 251 samples will be used, and for each of the Res-SincNet-FC based models, the hyperparameters that gave the best overall DER performance will be chosen. Table 4.31 shows the speaker recognition and speaker diarization performance of the different Res-SincNet-FC based models with sinc filter size equal to 251 samples, and the hyperparameters that give the best diarization performance for each model. The results in Table 4.31 are on the TIMIT dataset [71].

Table 4.31: The results of the models based on the Res-SincNet-FC architecture with sinc filter = 251 (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet-FC	-	32.51	1.52	35.67	34.43
AM-Res-SincNet-FC	$m = 0.5$	28.41	1.66	36.98	34.50
Arc-Res-SincNet-FC	$m = 0.5$	*27.17*	*1.44*	*35.16*	**31.97**
AdaCos-Res-SincNet-FC	Fixed s	27.83	1.52	36.24	*33.18*
PGL-Res-SincNet-FC	$\beta = 0.003$	33.65	1.95	34.93	32.20
MV-AM-Res-SincNet-FC	$m = 0.35, t = 0.02$	**26.14**	1.08	39.44	40.42
MV-Arc-Res-SincNet-FC	$m = 0.5, t = 0.02$	27.01	**1.01**	**34.44**	36.85
D-Res-SincNet-FC	$d = 0.9$	29.41	1.52	35.31	37.02

For each of the evaluation metrics in Table 4.31, the best values are in bold, the second-best values are underlined, while the third-best values are in italic. From Table 4.31, it can be seen that applying PCA improves the DER performance for most of the models. Furthermore, it can be observed that the Arc-Res-SincNet-FC gave the best DER performance compared to the other models, which was achieved when using PCA, also it achieved a relatively good FER, SER, and DER without PCA. The PGL-Res-SincNet-FC achieved the second-best DER performance with and without PCA. Additionally, from Table 4.31, it can be seen that the models that use the MV-Softmax loss [42], which are MV-AM-Res-SincNet-FC and MV-Arc-Res-SincNet-FC, gave the top two speaker recognition performance, where also the MV-Arc-Res-SincNet-FC achieved the best DER performance without PCA.

4.4.10 Results on the Librispeech Dataset

In this section, the Librispeech dataset [72], which is a relatively large dataset, is used to train and evaluate the models discussed in Section 4.4.9, where the hyperparameters that gave the best DER performance were chosen. This will help us study the effect of using a large dataset on the different performance measures across the different models. Table 4.32 shows the results of the different Res-SincNet-FC based models with the same settings as in Table 4.31, where the Librispeech dataset [72] is used.

Table 4.32: The results of the models based on the Res-SincNet-FC architecture with sinc filter = 251 on the Librispeech dataset (all values are in %)

Model	Hyperparameters	FER	SER	DER (w/o PCA)	DER (w/ PCA)
Res-SincNet-FC	-	13.64	0.85	26.77	28.27
AM-Res-SincNet-FC	$m = 0.5$	*13.00*	0.89	29.88	29.35
Arc-Res-SincNet-FC	$m = 0.5$	**12.00**	0.74	31.38	30.78
AdaCos-Res-SincNet-FC	Fixed s	13.47	1.05	30.05	31.32
PGL-Res-SincNet-FC	$\beta = 0.003$	13.41	0.81	*26.35*	27.21
MV-AM-Res-SincNet-FC	$m = 0.35, t = 0.02$	12.18	**0.68**	**26.05**	25.10
MV-Arc-Res-SincNet-FC	$m = 0.5, t = 0.02$	15.86	0.89	26.13	**24.75**
D-Res-SincNet-FC	$d = 0.9$	**12.00**	*0.75*	28.31	*26.14*

For each of the evaluation metrics in Table 4.32, the best values are in bold, the second-best values are underlined, while the third-best values are in italic. From Table 4.32, it can be seen that the speaker recognition and speaker diarization performance of all of the Res-SincNet-FC based models improved significantly compared to Table 4.15. This indicates that using a large dataset, such as the Librispeech dataset [72], with a Res-SincNet-FC based model will largely improve its performance. Moreover, it can be observed that using PCA improves the DER performance for several models.

From Table 4.32, it can be seen that the Arc-Res-SincNet-FC and the D-Res-SincNet-FC models achieved the best FER, and the second best and the third-best SER performance respectively. Also, the D-Res-SincNet-FC model achieved the third-best DER when PCA was applied. The MV-AM-Res-SincNet-FC model gave a particularly good performance, where it achieved the best SER and DER without PCA and the second-best FER and DER with PCA. Additionally, the MV-Arc-Res-SincNet-FC model achieved the best DER with PCA and the second-best DER without PCA.

In general, it can be seen that the models that employ the MV-Softmax loss [42], which are the MV-AM-Res-SincNet-FC and the MV-Arc-Res-SincNet-FC models, gave a better diarization performance compared to the other models. This shows that the MV-Softmax loss [42] is good for speaker embeddings extraction. This coincides with the findings in Section 4.2, where it was also found that the MV-Softmax loss gave the best DER when combined with the SincNet [1] architecture.

4.5 Analyzing and Visualizing the Diarization Output of the Best Model

From all the discussed models, which include the models based on: the SincNet architecture (discussed in Section 4.2), the Res-SincNet architecture (discussed in Section 4.3), and the Res-SincNet-FC architecture (discussed in Section 4.4), it can be seen that the proposed MV-AM-SincNet model gave the best DER performance. The MV-AM-SincNet model achieved a DER of 19.52%, when the Librispeech dataset [72] was used to train the MV-AM-SincNet model as speaker embeddings extractor, and when PCA was used. Thus, in this section, I will further analyze the MV-AM-SincNet diarization performance, by reporting its performance on the different meetings individually, and reporting its DER when the overlapped section are ignored (in all of the previously reported results overlapped speech was considered). Also, I will visualize the extracted d-vectors in the embedding space for all of the meetings. Finally, I will visualize and analyze the diarization output for one of the meetings.

Table 4.33 shows the DER performance of the MV-AM-SincNet model for the five meetings separately, where the DER with overlap and the DER without overlap are reported. It is worth noting that for both cases PCA is applied on the d-vectors.

From Table 4.33, it can be seen the DER without overlap is significantly better compared to the DER with overlap. This indicates that overlapped speech affects the diarization performance negatively, since the interfered audio from multiple speakers can

Table 4.33: The DER performance of the MV-AM-SincNet model on the different meetings (DER values are in %)

ID	Duration (min:sec)	Number of segments	DER (w/ overlap)	DER (w/o overlap)
IS1000a	26:22	389	14.44	9.70
IS1001a	15:09	184	19.87	14.08
IS1003b	27:26	424	9.30	4.43
IS1003d	35:00	783	34.31	26.74
IS1008d	24:40	341	8.12	1.66
Overall	-	-	19.52	11.66

cause confusion for the models when they are used to extract speaker embeddings from the overlapped segments.

Next, I will visualize the speaker embeddings extracted using the MV-SincNet model for the different meetings. T-SNE [84], which is a method used to visualize high-dimensional data, is used to visualize the extracted speaker embeddings (d-vectors). First, PCA is applied on the 2048 dimension d-vectors extracted from the last hidden layer to reduce their dimensions to 50. Then, length normalization is applied, which will give us the embeddings used to achieve the DER performances seen in Table 4.33.

4.5.1 Meeting IS1000a

This meeting has one female and three males, and it is around 27 minutes long. The meeting has 389 segments, where a segment is a duration of time spoken by a single speaker, and segments of different speakers can overlap. The MV-AM-SincNet model gave DER equal to 14.44% when overlapped segments were considered, and DER equal to 9.70%, when overlapped segments were ignored.

From Fig. 4.3, it can be seen that most of the speaker embeddings for the different speakers are nicely clustered. However, it can be seen some embeddings that are located in the wrong clusters. The embeddings that are in the wrong clusters could be extracted from short segments spoken by other speakers, where these short segments overlap with larger segments that correspond to the speaker of that cluster. Moreover, it can be observed that the speaker in red is the most dominant speaker (146 segments) in this meeting, and it can be seen that he has the most scattered embeddings, which could indicate that he has the

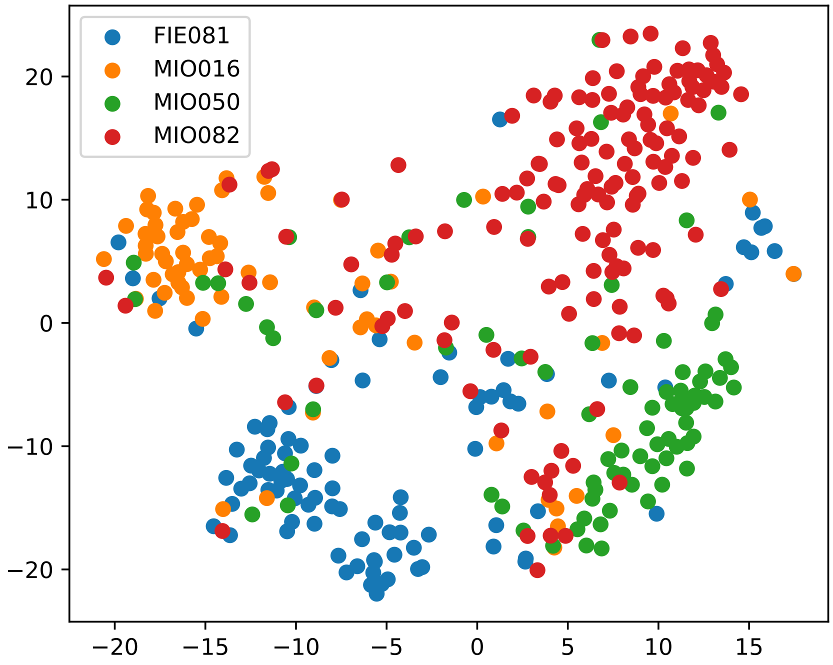

Figure 4.3: The TSNE plot of the speaker embeddings for meeting IS1000a

most overlapped speech with other speakers. The speaker in orange is the least dominant speaker (47 segments), while the speakers in blue (94 segments) and green (82 segments) contributed almost equally to the meeting.

4.5.2 Meeting IS1001a

This meeting has one female and three males, and it is around 16 minutes long, with 184 segments. The MV-AM-SincNet model gave DER equal to 19.87% when overlapped segments were considered, and DER equal to 14.08%, when overlapped segments were ignored.

From Fig. 4.4, it can be seen that there is a huge imbalance in the speakers' contribution to this meeting. it can be observed that the female speaker in blue has a significantly small number of segments (8 segments), and her embeddings overlap with embedding from the other speakers. Also, it can be seen that some of the embeddings of the speaker in green, who is the second least dominant speaker (32 segments), are clustered together, while some of his other embeddings are scattered. However, the speaker in red, who is the most

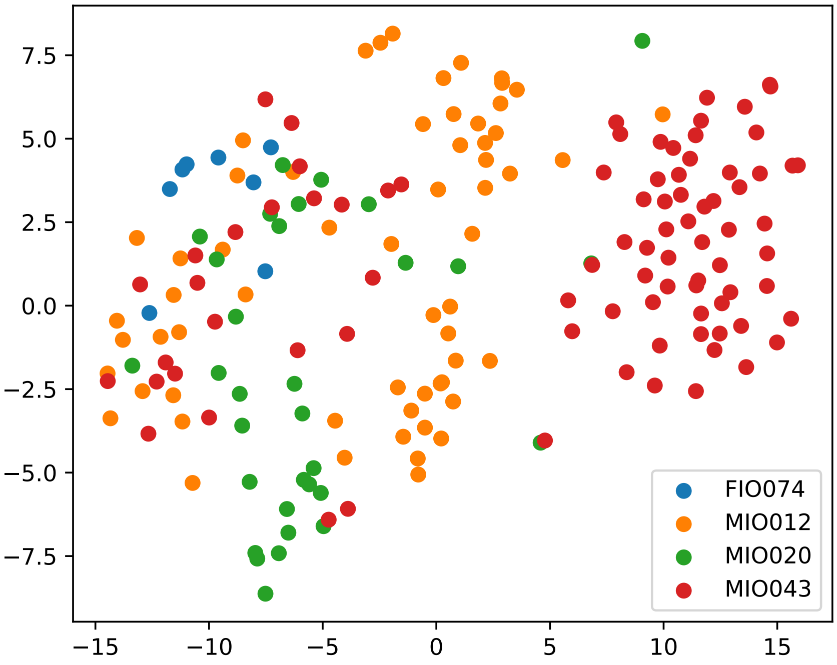

Figure 4.4: The TSNE plot of the speaker embeddings for meeting IS1001a

dominant speaker (84 segments), has the best clustered embeddings, followed by the orange speaker, who is the second most dominant speaker (60 embeddings). In general, it can be observed that the more dominant a speaker is, the better their embeddings are clustered.

4.5.3 Meeting IS1003b

This meeting has one female and three males, and it is around 27 minutes long, with 424 segments. The MV-AM-SincNet model performed really well on this meeting, where it gave DER equal to 9.30% when overlapped segments were considered, and DER equal to 4.43%, when overlapped segments were ignored.

From Fig. 4.5, it can be seen that the extracted speaker embeddings are highly discriminative, with very few embeddings scattered around, which can be due to overlapped speech. Additionally, it can be observed that the female speaker in blue, who is the most dominant speaker in this meeting (135 segments), is in a different region compared to the other male speakers, who are the speaker in orange (90 segments), the speaker in green (93 segments), and speaker in red (106 segments). Thus, it can be concluded that the

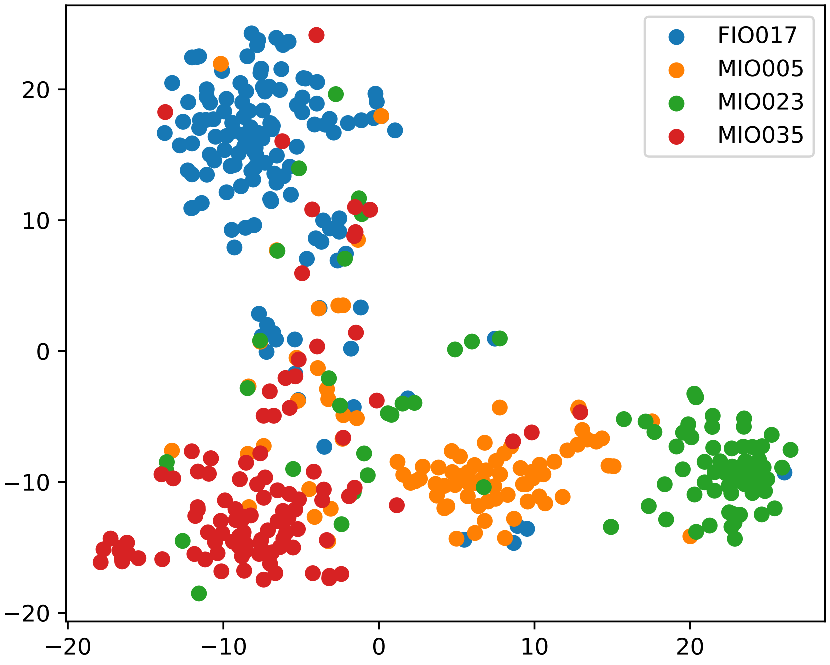

Figure 4.5: The TSNE plot of the speaker embeddings for meeting IS1003b

extracted speaker embeddings from a meeting having a balanced contribution from the different speakers, such as this meeting, and having a small number of overlapped speech, can result in discriminative embeddings, while also having the male and female speakers in different regions on the embeddings space.

4.5.4 Meeting IS1003d

This meeting has one female and three males, and it is around 36 minutes long, with 783 segments. The MV-AM-SincNet model performed the worst on this meeting, where it gave DER equal to 34.31% when overlapped segments were considered, and DER equal to 26.74%, when overlapped segments were ignored.

From Fig. 4.6, it can be seen that there are a lot of scattered embeddings, especially the embeddings of the speaker in orange (has 216 segments) and the speaker in red (has 189 segments). However, it can be seen the speaker embeddings corresponding to the speaker in blue (has 177 segments), and the speaker in green (has 201 segments) are more nicely clustered. The reason for the degraded performance on this meeting compared to

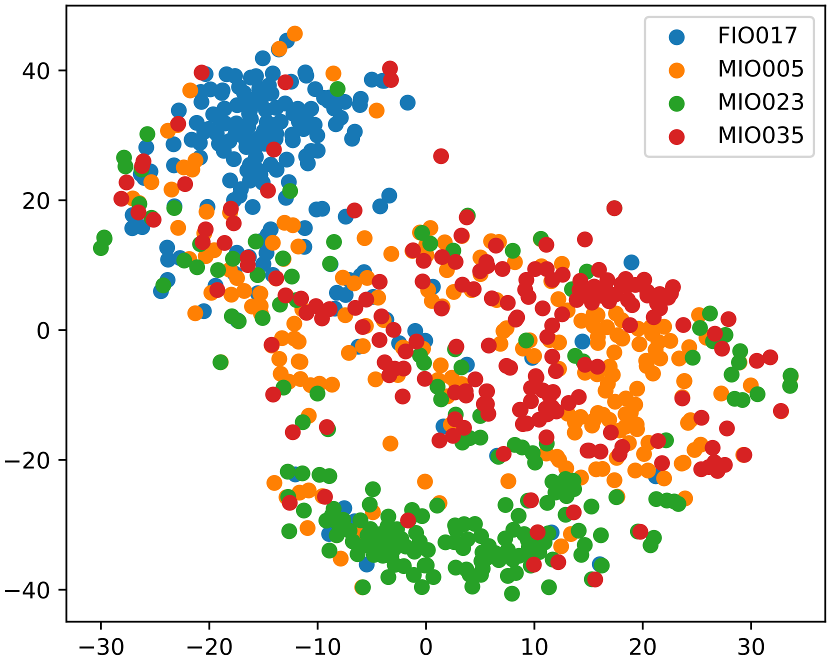

Figure 4.6: The TSNE plot of the speaker embeddings for meeting IS1003d

the previous meeting (IS1003b), is due to the fact that in this meeting, the team is having a discussion, which resulted in a huge number of overlapped segments. Thus, negatively impacting the diarization performance. Whereas, in meeting IS1003b, the different speakers were mostly presenting their ideas separately depending on their roles. Therefore, there is less overlapped speech in meeting IS1003b. Another indicator of the huge number of overlapped speech in meeting IS1003d compared to meeting IS1003b can be found in Table 4.33, where it can be seen that meeting IS1003d is around 7.5 minutes longer than meeting IS1003b, but it almost has double the number of segments. The relatively huge number of segments in meeting IS1003d can also indicate the presence of a huge number of quite small speaker segments. Therefore, it can be observed that even if the speakers have a balanced involvement in the meeting, having a huge amount of overlapped speech and numerous extremely small segments is bad for the diarization performance.

4.5.5 Meeting IS1008d

This meeting has two females and two males, and it is around 25 minutes long, with 341 segments. The MV-AM-SincNet model performed the best on this meeting, where it gave DER equal to 8.12% when overlapped segments were considered, and DER equal to 1.66%, when overlapped segments were ignored.

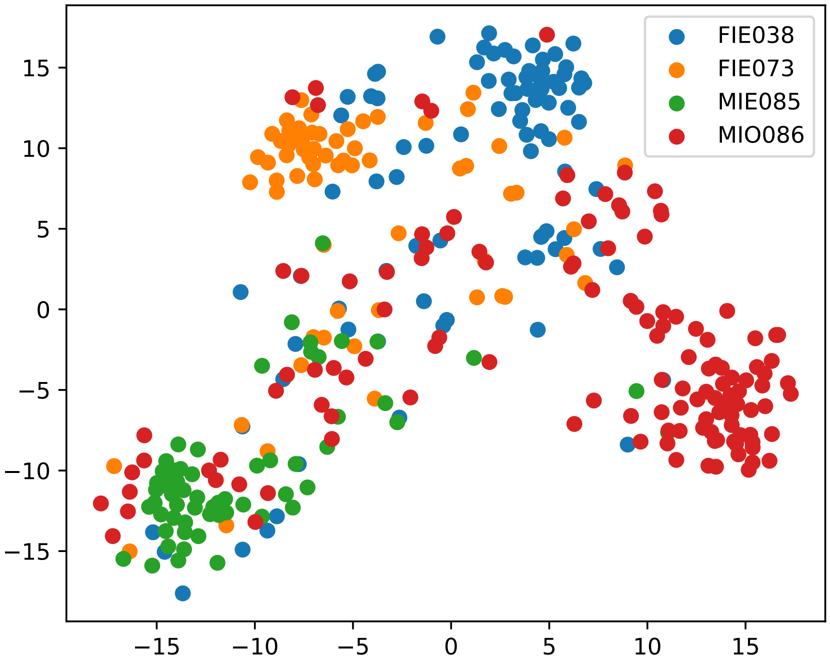

Figure 4.7: The TSNE plot of the speaker embeddings for meeting IS1008d

From Fig. 4.7, it can be seen that the speaker embeddings are highly discriminative (embeddings corresponding to the same speaker are close to each other, while speaker embeddings correspond corresponding to different speakers are far away from each other). However, there are some embeddings scattered around, which can be caused by overlapped speech, especially embeddings corresponding to the red speaker, who seems to have the most overlapped speech with other speakers. Moreover, it can be seen that the embeddings' clusters corresponding to the female speakers, who are the blue speaker (has 89 segments) and the orange speaker (has 65 segments), are close to each other in the embedding space, and they are in a different region compared to the embeddings' clusters corresponding to the male speakers, who are the green speaker (has 60 segments), and the red speaker (has 127 segments).

4.5.6 Visualizing the Diarization Output

To better understand the diarization output, the speaker turns in a random chunk of length 100 seconds from meeting IS1008d were visualized. The ground-truth chunk and diarization output of that chunk were visualized and compared. Fig. 4.8 shows the ground truth chunk, where it can be seen that it has segments corresponding to all the speakers with several overlaps.

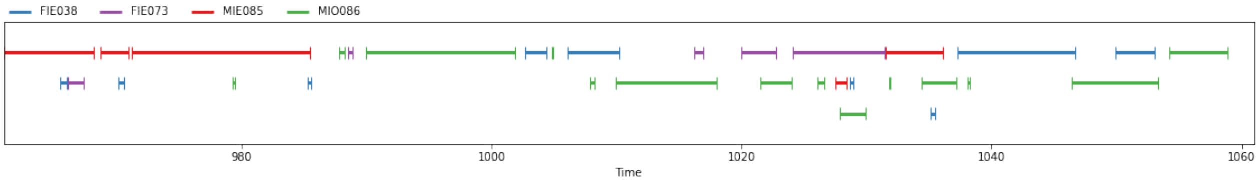

Figure 4.8: The ground-truth chunk from meeting IS1008d

The diarization output is depicted in Fig. 4.9, where it can be seen that it has the same segmentation as the ground-truth chunk, since the ground-truth segments are used in this thesis to avoid having segmentation errors as discussed in Section 4.1.4. Additionally, it can be seen that the diarization output has dummy speaker names.

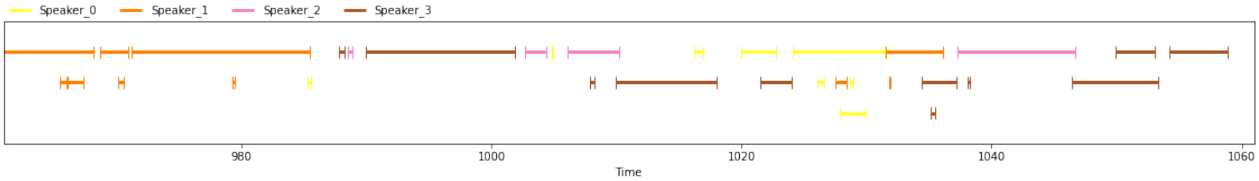

Figure 4.9: The diarization output of the chunk from meeting IS1008d

To facilitate a better comparison between the ground-truth chunk and the diarization output, the optimal mapping is found between the dummy speaker names in the diarization output and the speaker names in the ground-truth chunk. Fig. 4.10 shows the diarization output with optimal mapping, where it can be seen that the majority of the large segments got correctly attributed to the correct speakers. However, it can be observed that some short speaker segments that overlap with large segments of other speakers got incorrectly clustered. For example, at the begging of the chunk in Fig. 4.8, it can be seen that small segments of different speakers that overlap with larger segments corresponding to the red speaker. These small segments that overlap with the large segments got wrongly attributed to the speaker in red who spoke the large segments, as seen at the begging of Fig. 4.10. Furthermore, comparing Fig. 4.8 and Fig. 4.10, it can be observed that some relatively small segments got wrongly clustered.

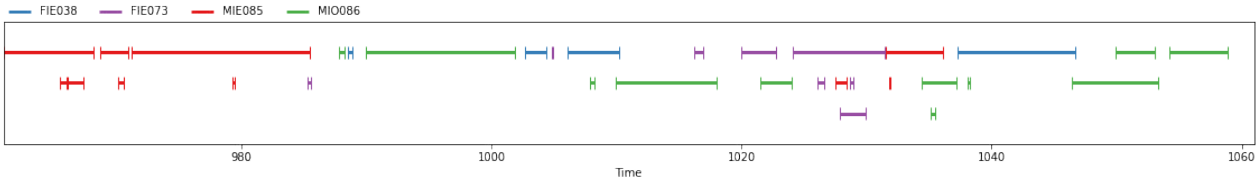

Figure 4.10: The diarization output with optimal speaker mapping of the chunk from meeting IS1008d

4.6 Summary

In this chapter, I concentrated on thoroughly discussing the experiments conducted in this thesis. Firstly, in Section 4.1 I went through the experimental setup, where I discussed the used datasets, which are the TIMIT dataset [71], the Librispeech dataset [72], and the AMI dataset [73]. Moreover, I explained the performance measures used to compare the different models in this thesis, which are the Frame Error Rate (FER), the Sentence Error Rate (SER), and the Diarization Error Rate (DER). I also went through the setup for the speaker recognition task and speaker diarization tasks, and I mentioned some major points regarding code implementation of the experiments.

Secondly, in Section 4.2 I reported, analyzed, and compared the results of the different models based on the SincNet architecture [1]. Different data augmentation methods were investigated, where it was found that using further augmentation methods is bad for the performance and is computationally expensive. Also, it was found that using a relatively large input window shift (50 ms) instead of the relatively small input window shift used in the original SincNet model [1] (10 ms) can reduce the computational requirements without having a major impact on the performance. Additionally, using a relatively large input window size gives a good FER performance, but it does not give a good SER and DER performance. On the other hand, using a relatively small input window size results in a good SER and DER performance, but with a bad FER performance. However, using a medium input window size (200 ms) gives good performance across the different performance measures.

After that, I investigated the proposed SincNet based models using different hyperparameters settings. Then, I compared the performance of the proposed models using the hyperparameters that gave the best DER performance with the existing SincNet based models. It was found that for most models there is a positive correlation between the SER performance and the DER performance, and applying PCA on the extracted d-vectors can result in a better DER performance. Furthermore, the models based on the MV-Softmax loss [42] performed the best on the different evaluation metrics compared to the other models, where the MV-AM-SincNet model achieved the best diarization performance. The models based on the MV-Softmax loss [42] were followed by the model based on the D-Softmax loss [57] (the D-SincNet model), which also performed well on the different performance measures.

Next, the size of the sinc filter was reduced from 251 to 31 samples which improved the FER performance, while also reducing the computational requirements. Most of the models gave around the same improved FER performance, which can be explained by a bottleneck caused by the SincNet architecture. However, reducing the size of the sinc filter significantly degrading the DER performance. Even after reducing the sinc filter size to 31 samples, the MV-AM-SincNet model gave the best DER performance. Afterward, I investigated the performance of the SincNet based models with the Librispeech dataset [72], which is a large dataset, where it helped in improving the speaker recognition and speaker Diarization of the SincNet based models. Again, the proposed MV-AM-SincNet gave the best diarization performance compared to the other models.

Thirdly, in Section 4.3 I reported, analyzed, and compared the results of the different models based on the Res-SincNet architecture [2], where it was found that non of the models that combine the Res-SincNet architecture [2] with the advanced softmax losses gave an improved performance compared to the Res-SincNet model [2], which uses the conventional softmax loss. The Res-SincNet architecture [2] is not suitable to be combined with the advanced softmax losses, since the global average pooling layer should be replaced with a fully connected layer to allow for the advanced losses to learn discriminative features in the penultimate layer.

Fourthly, in Section 4.4, I reported, analyzed, and compared the results of the different models based on the proposed Res-SincNet-FC architecture, which is based in the Res-SincNet architecture [2], where the global average pooling layer was replaced with the following layers (in their same mentioned order): batch normalization, dropout, fully connected, batch normalization, Leakey ReLU. Thus, the Res-SincNet-FC architecture is more suitable to be combined with the advanced softmax losses discussed in this thesis.

It was found that changing the architecture of the Res-SincNet model to Res-SincNet-

FC while using the conventional softmax loss is not enough to improve its performance. However, the models based on the Res-SincNet-FC architecture gave a significantly better performance when combined with the advanced softmax losses, compared to combining them with the Res-SincNet architecture [2]. Moreover, similar to the SincNet based models, increasing the size of the sinc filter for the Res-SincNet and the Res-SincNet-FC models is important to enhance their diarization performance. Increasing the sinc filter size of the Res-SincNet-FC resulted in better performance across all the reported performance measures, especially DER, compared to the Res-SincNet after also increasing its sinc filter size.

After that, I investigated the models based on the Res-SincNet-FC architecture with different hyperparameters settings, and using two sinc filter sizes, since it was found that using a relatively larger sinc filter size is good for speaker diarization, while a smaller sinc filter size is better for the speaker recognition. Thus, a sinc filter size of 89 samples was used to find the best performance on the speaker recognition task, and a sinc filter size of 251 samples to find the best performance on the speaker diarization task. It was found that for most models, reducing the dimensions of the extracted d-vectors using PCA is good for the diarization performance.

Then, I compared the different Res-SincNet-FC based models on the speaker recognition task, where a sinc filter size of 89 samples was used for all the models, and for each model, the hyperparameters that give the best speaker recognition performance were chosen. The D-Res-SincNet-FC and the AM-Res-SincNet-FC gave the best FER and SER respectively. It is worth noting that the speaker recognition performance of the Res-SincNet-FC based models is significantly better compared to the models based on the SincNet model with a sinc filter size of 31. Thus, the claim that the saturated FER performance achieved by most SincNet based models was due to the SincNet architecture, is true.

Afterward, I compared the different Res-SincNet-FC based models on the speaker diarization task, where a sinc filter size of 251 samples was used for all the models, and for each model, the hyperparameters that give the best speaker diarization performance were chosen. The Arc-Res-SincNet-FC gave the best DER performance, which was achieved after applying PCA. Next, I investigated the performance of the Res-SincNet-FC based models with the Librispeech dataset [72], which is a large dataset, where its usage greatly enhanced the speaker recognition and speaker diarization of the Res-SincNet-FC based models. The models based on the MV-Softmax loss [42] gave the best DER performance, where the MV-Arc-Res-SincNet-FC model achieved the best DER performance compare to all the other models. In general, it can be seen that in most cases, the models combined with the MV-Softmax losses [42] using either the SincNet architecture [1], or the proposed Res-SincNet-FC architecture, are always giving the best diarization performance. Therefore, it can be concluded that the MV-Softmax loss [42] has a superior performance at extracting discriminative embeddings compared to the other discussed losses. This also shows the robustness of the MV-Softmax loss [42], since it was mostly giving high diarization performance under different settings.

Finally, in Section 4.5, I focused on analyzing and visualizing the diarization results of the proposed MV-AM-SincNet model, which gave the best diarization performance compared to all the models discussed in this thesis. The best DER performance attained by the MV-AM-SincNet model was achieved when it was trained on the Librispeech dataset [72] after applying PCA on the extracted embeddings from the penultimate layer. The MV-AM-SincNet model performed exceptionally well on several meetings. Also, I reported and compared the DER performance when the overlapped regions are ignored and when they are included in the calculation of the DER, where the DER without overlap was better.

After visualizing the speaker embeddings extracted using the MV-AM-SincNet model for the different meetings, it was found that for most meetings the extracted embeddings were highly discriminative, where embeddings belonging to the same speakers were close to each other in the embedding space, and the different speaker clusters were far away from each other in the embedding space. This indicates the high capability of the MV-Softmax loss [42] at minimizing the inter-class variance and maximizing the inter-class variance. However, some embeddings were close to the wrong clusters, which can be the embeddings extracted from overlapped segments.

Moreover, it can be seen that having a huge imbalance in the contribution of the different speakers in a given meeting can negatively affect the speaker diarization performance. However, it can be observed that even if the speakers have a balanced involvement in the meeting, having a huge amount of overlapped speech and numerous extremely small segments is bad for the diarization performance. Furthermore, it can be seen that the MV-AM-SincNet model is somewhat able to distinguish between male speakers and female speakers, where it can be observed that clusters of the female speakers embeddings are close to each other in the embedding space, and the clusters of the male speaker embeddings are close to each other in the embedding space. Lastly, when visualizing the diarization output of a random chunk from one of the meetings, it was found that large segments got correctly attributed to their respective speakers. Whereas, relatively small segments, especially the ones that overlap with larger segments can sometimes be attributed to the wrong speakers.

Chapter 5

Conclusion and Future Work

In this chapter, I will conclude this thesis, and I will give some suggestions to further extend the work presented in this thesis.

5.1 Conclusion

In this thesis, various models that combine different SincNet architectures with distinct state-of-the-art loss functions were proposed. The proposed models are first trained to perform speaker recognition, then the trained models are used to extract speaker embeddings (d-vectors) in a speaker diarization system. Three architectures were considered, namely SincNet [1], Res-SincNet [2], and the proposed Res-SincNet-FC, where all three architectures are able to efficiently and effectively process raw input waveform, thanks to the sinc layer employed as the input layer for all three models. The use of raw input waveform stems from the desire to avoid using hand-crafted features, which may result in sub-optimal performance. The aforementioned architectures were combined with four state-of-the-art loss functions, which are AdaCos [55], PGL [56], MV-Softmax [42], and D-Softmax [57], where each of these losses improves on the softmax loss in a unique manner.

From the results, it was found that, generally, applying PCA on the extracted speaker embeddings, to reduce their dimensions, improved the diarization performance. Moreover, reducing the sinc filter size is crucial to obtain a low frame error rate (FER), but it hinders the diarization performance. However, a large sinc filter size is important to achieve a high diarization performance, but it causes the FER to increase. Furthermore, using a bigger dataset for training resulted in better speaker recognition and diarization performance.

It was found that the proposed MV-AM-SincNet model achieved the best diarization performance compared to all the models discussed in this thesis, where the MV-AM-SincNet model gave a significantly improved performance compared to the other models, while also achieving the best sentence error rate (SER). Also, the MV-AM-SincNet model consistently gave the best diarization performance across different datasets and different sinc filter sizes. Additionally, the MV-Arc-Res-SincNet-FC gave the best diarization performance compared to the other models that employ the same architecture (Res-SincNet-FC), but with other losses. Thus, it can be seen that the MV-Softmax loss [42] has a superior performance at extracting highly discriminative embeddings (d-vectors) compared to the other discussed losses. Moreover, this shows the robustness of the MV-Softmax loss [42], since it was mostly giving high diarization performance under different settings.

It was found that The Res-SincNet architecture [2] is not suitable to be combined with the loss functions discussed in this thesis, since the global average pooling layer does not offer the losses the flexibility to learn discriminative features in the penultimate layer. Thus, to solve this issue, the Res-SincNet-FC architecture was proposed, which replaces the average pooling layer with a fully connected layer, making it more suitable to be combined with the different losses. Furthermore, changing the architecture from SincNet [1] to Res-SincNet-FC resulted in a significantly lower FER, where the D-Res-SincNet-FC and Arc-Res-SincNet-FC gave the lowest FER compared to all the other models.

In addition, the diarization results of the MV-AM-SincNet model, which is the model that gave the best diarization performance, were analyzed and visualized. The diarization error rate (DER) with and without overlapped speech were compared, where the DER was lower when overlapped speech was ignored. The extracted speaker embeddings from the different meetings were visualized, where it was found that the embeddings are highly discriminative (i.e. speaker embeddings belonging to the same class are close to each other in the embeddings space, while embeddings from different classes are far away from each other). This indicates the high capability of the MV-Softmax loss [42] at maximizing both, intra-class compactness and inter-class separability. Also, for some meetings, the MV-AM-Softmax loss was able to differentiate between males and females in the embeddings space. However, some speaker embeddings were scattered around the embedding space, where these embeddings can be corresponding to overlapped speech. The diarization performance can be negatively affected if there is a large disparity in the number of segments spoken by different speakers. It was found that having a huge amount of overlapped speech and very small speaker segments can have a large impact on the diarization performance. Finally, visualizing the diarization output showed that large segments were correctly assigned to their correct speakers, while some small segments were assigned to the incorrect speakers, especially if these small segments overlap with large segments from other speakers.

5.2 Future Work

The work in this thesis can be further extended in several directions. For example, this thesis concentrated on improving the speaker embeddings extraction performance of the SincNet based architecture by employing different state-of-the-art loss functions; however, the performance can be also improved by focusing on changing the architecture.

For instance, in [85], the authors found that the diarization performance can be improved by using a recurrent convolutional neural network (RCNN) to extract embeddings, instead of a standard CNN, since recurrent layers are highly capable of aggregating and understanding temporal dependencies present in sequential and time-series data, such as audio signals. Thus, recurrent layers, such as LSTM [86] and GRU [87], can be incorporated with the SincNet based models discussed in this thesis to improve their diarization performance.

A second way to improve the architecture of the models discussed in this thesis is by replacing the convolutional layers with dilated convolutional layers. Dilated convolution increases the size of the receptive fields without requiring additional parameters, which will help reduce the number of layers compared to using standard convolution, hence reducing the overfitting problem caused by very deep networks [88]. Moreover, dilated convolution facilitates the learning of a wider temporal context compared to standard convolution [88].

Another suggestion to enhance the diarization performance of SincNet is by incorporating it with an end-to-end neural speaker diarization system, such as [89] or [90], where end-to-end systems are better than having separate modules (e.g. speaker modeling and clustering), since they directly optimize the diarization error rate (DER), resulting in a better performance. Also, end-to-end diarization systems can directly deal with overlapping speech, and they do not need non-overlapping speech for their training. In [89], the authors proposed a permutation-free loss function to achieve end-to-end speaker diarization, and in [90] the same authors improved on their end-to-end diarization system by introducing self-attention mechanism [91]. In both [89] and [90], the authors used hand-crafted features (log-Mel-filterbanks). Thus, they can be combined with the SincNet model [1] to allow them effectively and efficiently process raw input signals, which will enhance their performance and make them entirely end-to-end diarization systems, since the feature engineering step will be skipped.

Finally, the work in this thesis can be further extended by training the models discussed in this thesis using other famous datasets, such as VoxCeleb1 [92] and VoxCeleb2 [93]. Furthermore, the diarization performance of the models can be evaluated on other datasets, such as CALLHOME [94].

References

[1] M. Ravanelli and Y. Bengio, “Speaker recognition from raw waveform with SincNet,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, 2018.

[2] D. Oneață, L. Georgescu, H. Cucu, D. Burileanu, and C. Burileanu, “Revisiting SincNet: An evaluation of feature and network hyperparameters for speaker recognition,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2021.

[3] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, “Automatic speech recognition: a survey,” *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 9411–9457, 2021.

[4] S. Egan, “Speech & voice recognition software developers.” IBISWorld, May 2021.

[5] “Speech and voice recognition market worth $31.82 billion by 2025.” Grand View Research, May 2018.

[6] L. Fürer, N. Schenk, V. Roth, M. Steppan, K. Schmeck, and R. Zimmermann, “Supervised speaker diarization using random forests: a tool for psychotherapy process research,” *Frontiers in Psychology*, vol. 11, p. 1726, 2020.

[7] R. Yin, *Steps towards end-to-end neural speaker diarization*. PhD thesis, Paris Saclay, 2019.

[8] H. Dubey, A. Sangwan, and J. H. L. Hansen, “Transfer learning using raw waveform SincNet for robust speaker diarization,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6296–6300, 2019.

[9] J. A. Chagas Nunes, D. Macêdo, and C. Zanchettin, “Additive margin SincNet for speaker recognition,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5, 2019.

[10] L. Chowdhury, H. Zunair, and N. Mohammed, "Robust deep speaker recognition: Learning latent representation with joint angular margin loss," *Applied Sciences*, vol. 10, 10 2020.

[11] H. Dubey, *Deep Neural Networks and Model-Based Approaches for Robust Speaker Diarization in Naturalistic Audio Streams.* PhD thesis, The University of Texas at Dallas, 2019.

[12] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5239–5243, IEEE, 2018.

[13] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[14] L. Rabiner and R. Schafer, *Theory and applications of digital speech processing.* Prentice Hall Press, 2010.

[15] S. K. Mitra and Y. Kuo, *Digital signal processing: a computer-based approach*, vol. 2. McGraw-Hill New York, 2006.

[16] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with SincNet," *arXiv preprint arXiv:1811.09725*, 2018.

[17] D. Priyasad, T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Attention driven fusion for multi-modal emotion recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3227–3231, 2020.

[18] W. Wang, F. Seraj, N. Meratnia, and P. J. Havinga, "Speaker counting model based on transfer learning from SincNet bottleneck layer," in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–8, 2020.

[19] J. Fainberg, O. Klejch, E. Loweimi, P. Bell, and S. Renals, "Acoustic model adaptation from raw waveforms with SincNet," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 897–904, 2019.

[20] Y. Pan, V. S. Nallanthighal, D. Blackburn, H. Christensen, and A. Härmä, "Multitask estimation of age and cognitive decline from speech," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7258–7262, 2021.

[21] 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.

[22] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," 2019.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[24] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*, pp. 115–123, PMLR, 2013.

[25] S. Zhong, "Efficient online spherical k-means clustering," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 5, pp. 3180–3185, IEEE, 2005.

[26] R. Pilarczyk and W. Skarbek, "On intra-class variance for deep learning of classifiers," 2019.

[27] M. Wang and W. Deng, "Deep face recognition: A survey," 2020.

[28] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[29] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[30] Y. Sun, *Deep learning face representation by joint identification-verification.* The Chinese University of Hong Kong (Hong Kong), 2015.

[31] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2892–2900, 2015.

[32] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[33] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[34] E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of LFW benchmark or not?," *arXiv preprint arXiv:1501.04690*, 2015.

[35] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049–2058, 2015.

[36] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, "Template adaptation for face verification and identification," *Image and Vision Computing*, vol. 79, pp. 35–48, 2018.

[37] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*, pp. 499–515, Springer, 2016.

[38] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5409–5418, 2017.

[39] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 60–68, 2017.

[40] Y. Wu, H. Liu, J. Li, and Y. Fu, "Deep face recognition with center invariant loss," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pp. 408–414, 2017.

[41] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 302–309, IEEE, 2018.

[42] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, "Mis-classified vector guided softmax loss for face recognition," 2019.

[43] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," 2017.

[44] X. Liang, X. Wang, Z. Lei, S. Liao, and S. Z. Li, "Soft-margin softmax for deep classification," in *International Conference on Neural Information Processing*, pp. 413–421, Springer, 2017.

[45] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

[46] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large margin cosine loss for deep face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274, 2018.

[47] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, “NormFace: L2 hypersphere embedding for face verification,” in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

[48] B. Liu, W. Deng, Y. Zhong, M. Wang, J. Hu, X. Tao, and Y. Huang, “Fair loss: Margin-aware reinforcement learning for deep face recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10052–10061, 2019.

[49] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, “Adaptiveface: Adaptive margin and sampling for face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11947–11956, 2019.

[50] S. Zhou, C. Chen, G. Han, and X. Hou, “Double additive margin softmax loss for face recognition,” *Applied Sciences*, vol. 10, p. 60, 12 2019.

[51] D. Zhou, L. Wang, K. A. Lee, Y. Wu, M. Liu, J. Dang, and J. Wei, “Dynamic margin softmax loss for speaker verification.,” in *INTERSPEECH*, pp. 3800–3804, 2020.

[52] Y.-Q. Yu, L. Fan, and W.-J. Li, “Ensemble additive margin softmax for speaker verification,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6046–6050, IEEE, 2019.

[53] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, “Measuring statistical dependence with Hilbert-Schmidt norms,” in *International conference on algorithmic learning theory*, pp. 63–77, Springer, 2005.

[54] J. Sun, W. Yang, R. Gao, J.-H. Xue, and Q. Liao, “Inter-class angular margin loss for face recognition,” *Signal Processing: Image Communication*, vol. 80, p. 115636, 2020.

[55] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, “AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations,” 2019.

[56] Y. Qin, C. Yan, G. Liu, Z. Li, and C. Jiang, “Pairwise Gaussian loss for convolutional neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6324–6333, 2020.

[57] L. He, Z. Wang, Y. Li, and S. Wang, “Softmax dissection: Towards understanding intra- and inter-class objective for embedding learning,” 2020.

[58] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization.,” *Journal of machine learning research*, vol. 13, no. 2, 2012.

[59] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in *International conference on learning and intelligent optimization*, pp. 507–523, Springer, 2011.

[60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[61] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*, vol. 2, Lille, 2015.

[62] Y. Sun, *Deep learning face representation by joint identification-verification.* The Chinese University of Hong Kong (Hong Kong), 2015.

[63] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[64] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[66] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2017.

[67] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[68] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[71] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, p. 27403, 1993.

[72] 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)*, pp. 5206–5210, 2015.

[73] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th international conference on methods and techniques in behavioral research*, vol. 88, p. 100, Citeseer, 2005.

[74] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[75] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.

[76] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[77] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[78] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.

[79] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[80] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[81] H. Bredin, "pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, (Stockholm, Sweden), August 2017.

[82] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks," 2020.

[83] L. Nanni, G. Maguolo, and M. Paci, "Data augmentation approaches for improving animal audio classification," 2020.

[84] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[85] P. Cyrta, T. Trzciński, and W. Stokowiec, "Speaker diarization using deep recurrent convolutional neural networks for speaker embeddings," in *International Conference on Information Systems Architecture and Technology*, pp. 107–117, Springer, 2017.

[86] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[87] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[88] Y. Li, M. Liu, K. Drossos, and T. Virtanen, "Sound event detection via dilated convolutional recurrent neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 286–290, IEEE, 2020.

[89] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *arXiv preprint arXiv:1909.05952*, 2019.

[90] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 296–303, IEEE, 2019.

[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[92] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," *Interspeech 2017*, Aug 2017.

[93] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," *Interspeech 2018*, Sep 2018.

[94] G. Z. Alexandra Canavan, David Graff, "CALLHOME american english speech LDC97S42." Philadelphia: Linguistic Data Consortium, 1997.