

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Priit Käär 211795IAPM

**WEAKLY SUPERVISED SPEAKER IDENTIFICATION  
SYSTEM IMPLEMENTATION BASED ON ESTONIAN  
PUBLIC FIGURES**

Master's Thesis

Supervisor: Tanel Alumäe  
Tenured Associate Professor

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Priit Käär 211795IAPM

**KAUDSELT JUHENDATUD KÕNELEJATUVASTUSE  
SÜSTEEMI IMPLEMENTATSIOON EESTI AVALIKU ELU  
TEGELASTE NÄITEL**

Magistritöö

Juhendaja: Tanel Alumäe  
Kaasprofessor Tenuuris

Tallinn 2023

## **Author's Declaration of Originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Priit Käär

07.05.2023

# Abstract

Speaker identification is a task of assigning speaker identities to speech segments in an audio recording. Typically, training speaker identification systems requires hand-labelling the data by experts, which is expensive and time-consuming. Weakly supervised learning solves this problem by using only recording-level labels. These labels can be acquired from existing metadata (e.g. descriptions) faster and at lower costs compared to human labour.

The thesis studies recent advances in machine learning methods for speaker identification. An existing implementation of a weakly supervised speaker identification system is used as a baseline. Several improvements proposed from recent years are implemented, including for example data augmentation techniques, using and fine-tuning a state-of-the-art x-vector embeddings extractor. The improvements are validated by measuring precision and recall on several test sets from different domains.

The experiments show that the improved weakly supervised speaker identification system can maintain high precision even for out-of-domain datasets and at the same time increase recall compared to the baseline system. That means the improved system is more robust to channel variances and speakers with similar voice characteristics, therefore high precision is maintained as the data evolves. At the same time the improved system can recognize more speakers that appear in the training data.

The thesis is written in English and is 49 pages long, including 7 chapters, 16 figures and 20 tables.

## **Annotatsioon**

### **Kaudselt Juhendatud Kõnelejatuvastuse Süsteemi Implementatsioon Eesti Avaliku Elu Tegelaste Näitel**

Kõnelejatuvastus on ülesanne, mis määrab kõneleja identiteedi igale kõne segmendile helifailis. Tavaliselt vajab kõnelejatuvastuse süsteemi treenimine ekspertide poolt käsitsi märgendatud andmeid, mida on kallid ja ajakulukas koguda. Kaudselt juhendatud õppimine lahendab selle probleemi, kasutades ainult salvestuse tasemel märgendusi juba olemasolevatest metaandmetes (näiteks kirjeldus). Selliseid andmeid on oluliselt kiirem ja odavam koguda.

Magistritöö eesmärgiks on uurida hiljutisi edusamme masinõppe meetodites kõnelejatuvastuse läbiviimiseks. Alusena on kasutatud olemasolevat kaudselt juhendatud kõnelejatuvastuse süsteemi, samuti on rakendatud mitmeid viimastel aastatel väljapakutud täiustusi. Need täiustused sisaldavad näiteks andmete paljundamist, ette-treenitud kõneleja x-vektorite eraldaja kasutamist ja kohendamist. Süsteemi täiustused valideeritakse mõõtes täpsust ja saagist erinevate valdkondade test andmete peal.

Katsed näitavad, et täiustatud kaudselt juhendatud kõnelejatuvastuse süsteem suudab säilitada kõrge täpsuse ka eri valdkondade andmete peal, mida mudel ei ole varem näinud, samas tõsta ka saagist. See tähendab, et täiustatud mudel suudab täpsemini tuvastada sarnase kõnestiiliga isikuid üle erinevate kanalite ning tuvastada rohkem treeningandmetes kohatud isikuid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 49 leheküljel, 7 peatükki, 16 joonist, 20 tabelit.

## List of Abbreviations and Terms

BN	Batch Normalization
Conv1D	1-Dimensional Convolutional Layer
DCT	Discrete Cosine Transform
DNN	Deep Neural Network
ECAPA	Emphasized Channel Attention, Propagation and Aggregation
ERR	Estonian Public Broadcasting
FC	Fully-Connected Layer
GMM	Gaussian Mixture Model
JFA	Joint Factor Analysis
MLM	Masked Language Model
ReLU	Rectified Linear Unit
SE	Squeeze-Excitation
TDNN	Time Delay Neural Network
UBM	Universal Background Model
MFCC	Mel-Frequency Cepstral Coefficient

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Background Theory</b>	<b>10</b>
2.1	Speaker Recognition	10
2.1.1	Speaker Diarization	10
2.1.2	Speaker Identification	10
2.2	Weakly Supervised Learning	11
2.2.1	Overview	11
2.2.2	Weakly Supervised Speaker Identification	12
2.2.3	Method	13
2.3	Mel-Frequency Cepstral Coefficients	15
2.4	Embeddings	17
2.4.1	I-vectors	17
2.4.2	X-vectors	18
2.4.3	ECAPA-TDNN	18
<b>3</b>	<b>Data</b>	<b>22</b>
3.1	Sources	22
3.1.1	Estonian Public Broadcasting Archive	22
3.1.2	Soundcloud	22
3.2	Data Acquisition	23
3.3	Statistics	24
3.3.1	Total number of shows used	24
3.3.2	Total duration of recordings	24
3.3.3	Number of recordings annually	25
3.3.4	Average number of speaker occurrences per show annually	25
3.3.5	Frequency Rank vs. Appearances	26
3.3.6	Most Frequent Speakers per Show	27
<b>4</b>	<b>Experimental Setup</b>	<b>30</b>
4.1	Overview	30
4.2	Model Architecture	30
4.3	KaldiDataset	31
4.4	Training	31
4.5	Inference	32

<b>5</b>	<b>Improvements</b>	<b>33</b>
5.1	Adjusting the Posterior Probabilities	33
5.2	Data Augmentation	34
5.3	Pre-Trained Models	34
5.3.1	Kaldi’s TDNN-UBM I-vectors	34
5.3.2	Speechbrain’s ECAPA-TDNN X-vectors	35
5.4	Fine-Tuning ECAPA-TDNN	35
5.5	SpecAugment	36
<b>6</b>	<b>Evaluation</b>	<b>38</b>
6.1	Baseline	38
6.2	Larger and More Recent Dataset	38
6.3	Adjusting the Posterior Probabilities	39
6.4	Speechbrain’s ECAPA-TDNN	39
6.5	Data Augmentation	40
6.6	Fine-Tuned ECAPA-TDNN	41
6.7	SpecAugment	41
6.8	Threshold Tuning	42
6.9	Summary	43
<b>7</b>	<b>Summary</b>	<b>45</b>
	<b>References</b>	<b>46</b>
	<b>Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis</b>	<b>49</b>



## List of Figures

1	The process of speaker diarization. A typical speaker diarization system consists of a speech detection stage, a segmentation stage, and a clustering stage [3]. . . . .	10
2	Speaker recognition fundamental tasks (verification and identification) [4].	11
3	Speaker verification system with frontend and backend diversity [5]. . . .	12
4	Time-based annotations required for supervised learning in speaker identification systems. . . . .	13
5	Recording-level labels required for weakly supervised learning in speaker identification systems. . . . .	13
6	Overview of a weakly supervised speaker identification system proposed by Martin Karu and Tanel Alumäe. [1] . . . . .	14
7	Forms of an audio signal during the MFCCs calculation [8]. . . . .	16
8	The i-vector extraction framework [10]. . . . .	17
9	Architecture of the x-vector system [13]. . . . .	19
10	Network topology of the ECAPA-TDNN [14]. . . . .	20
11	Description of an episode in "Arvamusfestival". . . . .	23
12	Number of recordings per show annually. . . . .	25
13	Average number of speaker occurrences in a recording per show annually.	26
14	Average number of speaker occurrences per show. . . . .	27
15	Spec-Augment applied on an audio spectrogram [23]. . . . .	37
16	Precision vs. recall with thresholds at 95 % precision across datasets. . . .	43

## List of Tables

1	Total number of recordings per show and dataset. . . . .	24
2	Total duration of recordings per show and dataset in minutes. . . . .	24
3	Top 10 most frequent speakers in "Uudised". . . . .	28
4	Top 10 most frequent speakers in "Päevakaja". . . . .	28
5	Top 10 most frequent speakers in "Reporteritund". . . . .	28
6	Top 10 most frequent speakers in TV. . . . .	29
7	Top 10 most frequent speakers in "Arvamusfestival". . . . .	29
8	Baseline model performance metrics. . . . .	38
9	Re-trained model performance metrics on a larger dataset. . . . .	39
10	Re-trained model performance metrics with adjusted posterior probabilities. . . . .	39
11	X-vector-based model performance metrics. . . . .	40
12	X-vector-based model performance metrics with posterior probability adjustment. . . . .	40
13	Model performance metrics with data augmentation. . . . .	40
14	Model performance metrics with data augmentation and posterior probability adjustment. . . . .	41
15	Model performance metrics with fine-tuning. . . . .	41
16	Model performance metrics with fine-tuning and posterior probability adjustment. . . . .	41
17	Model performance metrics with SpecAugment. . . . .	42
18	Model performance metrics with SpecAugment and with posterior probability adjustment. . . . .	42
19	Relative F1 measure baseline improvements across datasets. . . . .	42
20	Relative precision and recall improvements across datasets. . . . .	44

# 1. Introduction

Speaker identification is a task of recognizing the identity of a speaker based on his/her voice. This task has a wide range of applications, from forensic investigations to voice-based authentication systems. Common speaker identification systems rely on supervised learning, which require large amount of discretely labelled data. However, collecting discretely labelled data is a time-consuming and expensive process.

To address this issue, previous research has proposed weakly supervised approaches to speaker identification systems. In particular, the thesis of Martin Karu and Tanel Alumäe in 2018 proposed a method for speaker identification using audio recordings with only recording level labels [1].

The thesis aims to improve the aforementioned existing weakly supervised speaker identification system. Specifically, a larger and more recent dataset from the Estonian Public Broadcasting archive is used to train the model. Furthermore, a more recent pre-trained model is used for calculating speaker embeddings for input to the speaker identification model. The pre-trained speaker embedding model is also fine-tuned on raw audio files as an input instead of using static pre-calculated speaker embeddings. In addition to the in-domain recordings from the Estonian Public Broadcasting archive, the improved versions are also evaluated using out-of-domain audio recordings (e.g. "Arvamusfestival"). The hypothesis is that at least 90 % precision can be achieved in identifying corresponding speakers on both in-domain and out-of-domain datasets and at the same time recall is increased compared to the baseline model by using these techniques.

## 2. Background Theory

### 2.1 Speaker Recognition

Speaker recognition in general consists of several sub-tasks, including speaker diarization, verification, and identification.

#### 2.1.1 Speaker Diarization

Speaker diarization is a process of partitioning an audio stream into homogeneous speaker segments and determining which segments are uttered by the same speaker [2]. This process is often categorised further into speech detection, speaker segmentation and speaker clustering stages. Speaker diarization is typically a preliminary step to speaker verification or identification process as it reduces the complexity of the process down to a single speaker. The process of a typical speaker diarization system is depicted in Figure 1.



Figure 1. The process of speaker diarization. A typical speaker diarization system consists of a speech detection stage, a segmentation stage, and a clustering stage [3].

#### 2.1.2 Speaker Identification

Speaker verification and speaker identification are two very similar areas of research. Speaker verification is a process of deciding whether an unknown speaker segment belongs to a specific reference speaker or not [2]. Latter has only two possible outcomes - whether to accept the reference speaker or reject the impostor. Speaker verification is commonly used in forensic investigation and voice-based authentication systems. Speaker identification differs from speaker verification only by deciding between multiple reference speakers instead of one [2]. Speaker identification systems are often seen in automatic transcription systems. Figure 2 depicts the difference in speaker identification and verification systems.

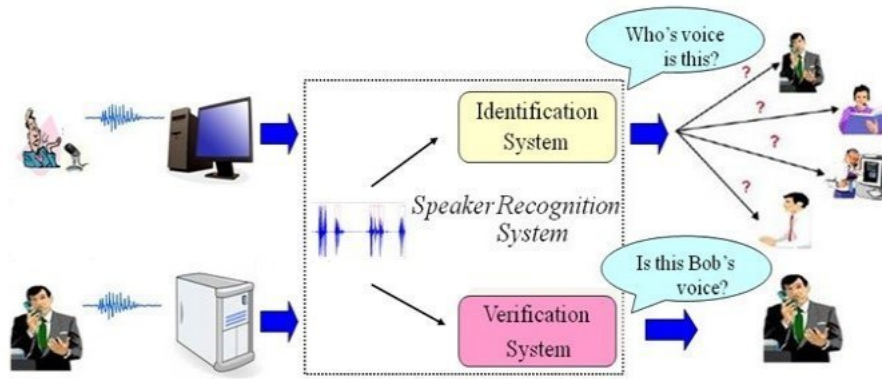


Figure 2. Speaker recognition fundamental tasks (verification and identification) [4].

A speaker identification system typically uses speaker diarization as a preliminary step to reduce the complexity of the problem down to a single speaker.

Speaker identification process typically consist of two phases: enrolment and recognition, also known as training and testing phases [2].

In most modern speaker identification systems, a model is trained for each speaker separately in the enrolment phase. Each speaker in the training set will be associated with a feature vector called embedding. The embedding is a compact representation of the unique characteristics of a speaker's voice. In the recognition phase, an unknown speaker segment is provided to a classifier that decides which speaker in the training set is the most similar to the unknown speaker.

In some literature, speaker recognition models are divided into two sections: frontend and backend. Frontend is applied both in the enrolment and the recognition phase and is responsible for extracting the features (e.g. MFCCs) from a raw audio waveform, that are later used for speaker modelling. Backend is responsible for the speaker modelling and either identifying or verifying the speaker identity by comparing the embeddings from the known speakers to the embedding from the unknown speaker's speech signal (see Figure 3).

## 2.2 Weakly Supervised Learning

### 2.2.1 Overview

Machine learning is typically divided into supervised and unsupervised learning. Unsupervised learning does not require any labels on the data that is used to train a machine learning model. Usually these methods are used for solving clustering, association pattern mining,

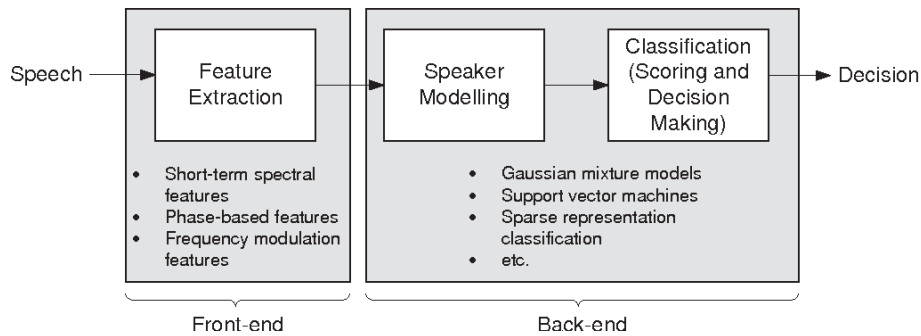


Figure 3. Speaker verification system with frontend and backend diversity [5].

and dimensionality reduction problems. Supervised learning requires labels on the data that is used to train a model. These methods are typically used for solving classification and regression problems.

Speaker identification is a type of multi-label classification problem, therefore supervised learning is the most suitable approach for solving the problem. However, collecting large amount of labelled data for a speaker identification task is time-consuming and expensive. Fortunately, weakly supervised learning can be used to overcome this issue.

Weakly supervised learning refers to machine learning methods where datasets are used with partially labelled data. Many types of weak supervision exist, including:

- Candidate labels: Multiple labels are assigned to each training sample, while only one of them is correct.
- Probabilistic labels: Every label is assigned to every training sample with a given probability.
- Incomplete labelling: Each training sample belongs to multiple classes, but only a partial set of the classes are labelled for each sample.
- Crowd annotation: Labelling is done by non-expert and cheap labour, thus the labels are not very trustworthy.
- Label proportions: The proportions of the labels in a set of instances are known, but not which ones precisely correspond to each label.

### 2.2.2 Weakly Supervised Speaker Identification

Speaker identification systems aim to identify a person based on his/her voice. Training a speaker identification model usually requires time-based annotations on the training data. However, it is difficult to cover a wide range of speakers (e.g. thousands of politicians for media monitoring purposes) and it is difficult to keep it up-to-date as it requires

hand-labelling additional data. An example of time-based annotation is seen in Figure 4.

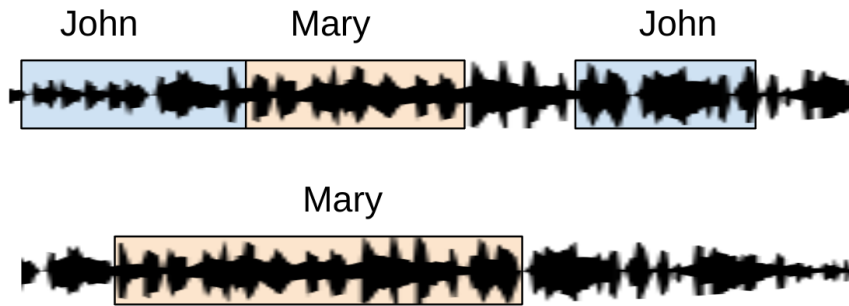


Figure 4. Time-based annotations required for supervised learning in speaker identification systems.

Weakly supervised speaker identification system however requires only recording-level labels. That means only the set of speakers who appear somewhere in the recording are known, therefore time-based annotation is not required (see Figure 5). This simplifies the data acquisition process, since there is a lot more data already available on the internet that already include this kind of metadata (e.g. textual descriptions describing the contents and speakers of a radio show).



Figure 5. Recording-level labels required for weakly supervised learning in speaker identification systems.

### 2.2.3 Method

In 2017, Martin Karu and Tanel Alumäe proposed a weakly supervised method for speaker identification (see Figure 6). For training, a list of audio files is used only with the corresponding sets of speakers. Training the system requires speaker diarization, which outputs a list of automatically segmented audio files, where each utterance belongs to an unknown homogeneous speaker. Each segment is then transformed into a speaker embedding. Since it is unknown which embeddings correspond to which speakers, a deep

neural network is trained that maps speaker embeddings to the speakers known from the training data, using a special cost function. [1]

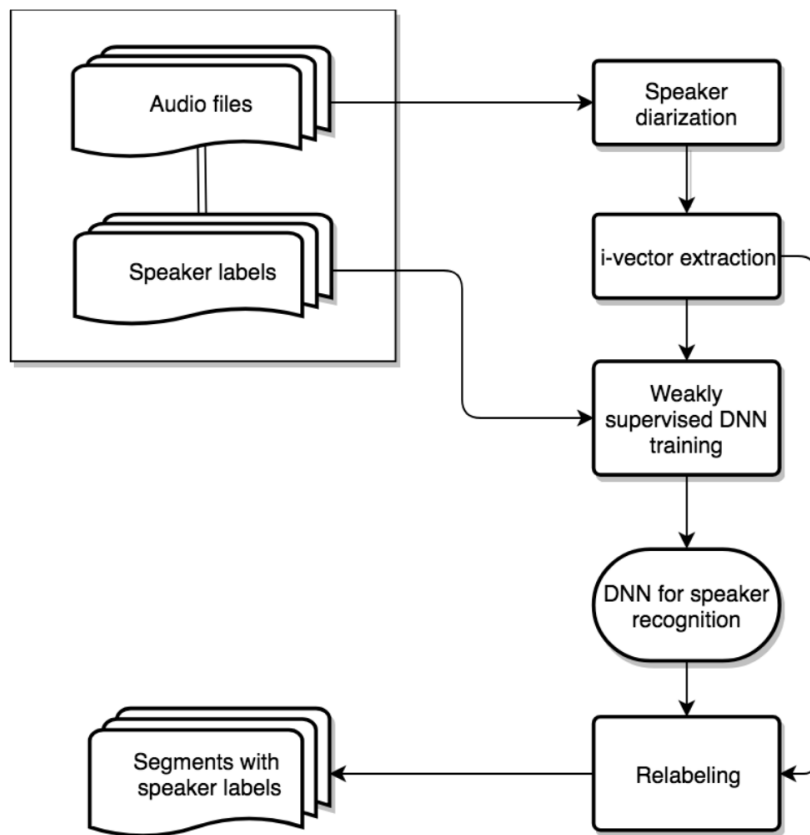


Figure 6. Overview of a weakly supervised speaker identification system proposed by Martin Karu and Tanel Alumäe. [1]

The weakly supervised DNN uses a set of recordings for training, each of which consists of a set of persons-of-interest that speak there, and a set of embeddings. However, it is unknown which embeddings correspond to which speakers. Also, there could be more embeddings compared to the number of speaker labels.

The weakly supervised DNN is used to compute posterior probabilities for all embeddings of the diarized speakers in a recording. These posterior probabilities are then averaged (predicted average), which results in a probability distribution across speakers. An expected average, described in Equation 2.1, is also calculated for the special cost function. Finally, Kullback-Leibler divergence is calculated between the predicted and expected average distributions. Until the model has not converged during training, the DNN gradients and model weights are recalculated during back-propagation, and the process is repeated. [1]



$$p_n(y_i) = \begin{cases} \frac{1}{|X_n|} & \text{if } y_i \in Y_n \\ \max(0, 1 - \frac{|Y_n|}{|X_n|}) & \text{if } y_i = \langle \text{unk} \rangle \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

### Intuition

Assume there are five speaker embeddings (five diarized speakers) and two speaker labels (John and Mary) from a single recording.  $X_n = 5$  is the number of detected speakers. For John and Mary, the expected average posterior probability is  $\frac{1}{5}$ . For other known speakers, the expected average posterior is zero. For the unknown speaker label (sort of background model), the expected average is  $1 - \frac{2}{5} = \frac{3}{5}$ .

Therefore, the model is encouraged to produce non-zero average posterior speaker A only if speaker A occurs in the corresponding recording. For all other recordings, the model is trained to produce a zero average posterior for speaker A.

### Limitations

The described method also introduces some limitations:

- Each person-of-interest should appear in several recordings (the more, the better).
- Two persons should not always appear in the same recording (otherwise there is no way to distinguish which embedding belongs to which speaker).

## 2.3 Mel-Frequency Cepstral Coefficients

The human auditory system is more sensitive to some frequency bands than others. This sensitivity is not linear across the frequency range. MFCCs take this into account by using the Mel-scale, which is a logarithmic scale that maps frequencies from Hz to a perceptually relevant scale. [6]

The Mel-scale is based on the observation that humans perceive differences in low-frequency sounds more easily than differences in high-frequency sounds. Therefore, MFCCs use more filter banks to cover the lower frequency range, and fewer filter banks for the higher frequency range. [6]

MFCCs also involve taking the logarithm of the magnitude of the speech spectrum, which compresses the dynamic range of the signal and makes it easier to work with. After

applying the Mel-scale, the logarithmic transformation, and other additional processing steps (e.g. applying a Discrete Cosine Transform), the resulting MFCCs provide a compact representation of the spectral characteristics of a speech signal. [7]

Calculating MFCCs consists of the following steps:

1. Pre-emphasis: The speech signal is passed through a high-pass filter to emphasize higher frequencies [7].
2. Framing: The pre-emphasized signal is divided into overlapping frames, typically with 20-40ms durations[7].
3. Windowing: A windowing function (e.g. Hamming) is applied to each frame to smoothen the frame boundaries [7].
4. Fast Fourier Transform: The Fourier transform is applied to each frame to convert it from time domain to the frequency domain. The frequency domain can be visualized as a spectrogram over the time-domain [7].
5. Mel-filterbank: A bank of filters, spaced according to the Mel-scale, is applied to the magnitude spectrum of each frame. This results in the energy within each of the frequency bands.
6. Discrete Cosine Transform: The DCT is applied to the logarithm of the energies from the filterbanks to obtain a set of cepstral coefficients [7].
7. Finally, the first few coefficients are discarded, since they represent the overall energy / gain of the signal. The rest of the coefficients (2-13) are kept as MFCCs [7].

Different forms of the audio signal during the calculation process are depicted in Figure 7.

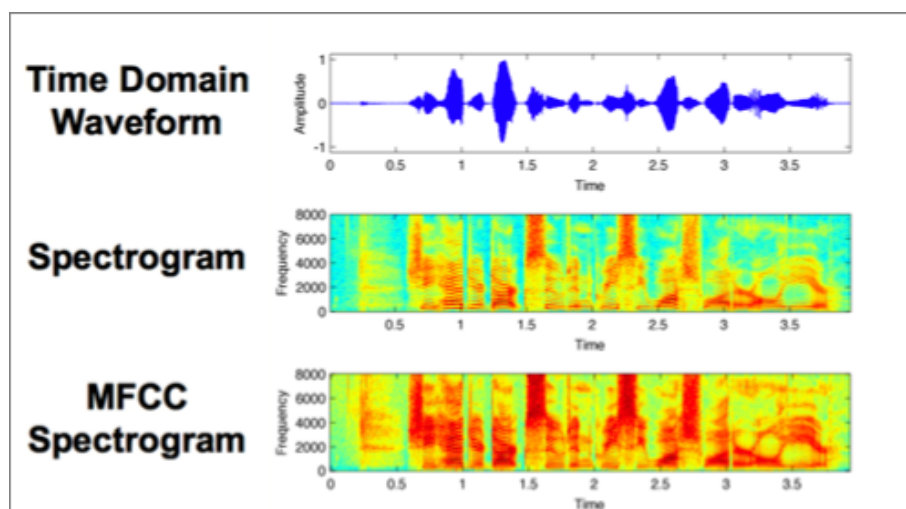


Figure 7. Forms of an audio signal during the MFCCs calculation [8].

## 2.4 Embeddings

Extracting features that represent the most discriminative features of a speaker's voice is the key problem in speaker identification systems. This section describes different algorithms used to calculate speaker embeddings and how they have improved over time.

### 2.4.1 I-vectors

For years GMM-UBM (Gaussian Mixture Model - Universal Background Model) was the state-of-the-art method used for speaker recognition tasks. In essence, it follows a standard speaker identification process. Firstly, a Gaussian mixture model is trained across all the speakers in the training set. Secondly, for each speaker in the training set, the means of the Gaussian mixture model are adjusted. As a result, the adapted mixture components from the GMM are used as a speaker embedding, also referred to as a GMM supervector [9].

However, the traditional GMM-UBM method does not perform well due to speaker and channel variability (noise, echoes, distortions from the microphone; pitch, gender, dialect of the speaker; etc.). In 2011, i-vectors were introduced to overcome the speaker and channel variability using joint factor analysis (JFA) on top of the traditional GMM-UBM approach [9].

I-vectors output a compact representation of the unique speaker's voice characteristics (embeddings) together with the total variability matrix to overcome the previously mentioned speaker and channel variability issues [9].

The i-vector extraction framework is depicted in Figure 8.

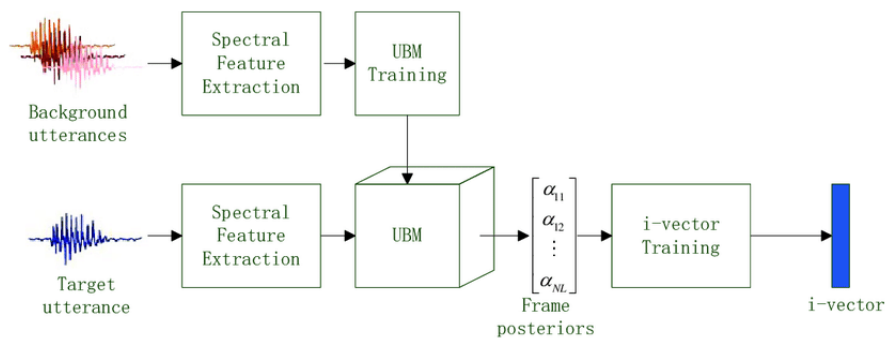


Figure 8. The i-vector extraction framework [10].

## 2.4.2 X-vectors

With the popularization of deep neural networks, alternative approaches to the traditional GMM-UBM method were proposed. In 2018, David Snyder et al. proposed a method to map variable-length utterances to fixed-dimensional embeddings, called x-vectors, using deep neural networks [11].

The x-vector system consists of five frame-level layers, one statistical pooling layer, two segment-level layers, and a soft-max layer. In the experimental setup, the input used 30-dimensional MFCC features, extracted from 25ms audio signal frames, mean-normalized over a sliding window of up to 3 seconds [12].

Suppose  $t$  is the current time step. Frames from  $(t-2)$  to  $(t+2)$  are spliced at the first layer. On the second frame layer, the output of the first layer is spliced at time steps  $(t-2)$  to  $t$  and  $t$  to  $(t+2)$ . The third frame layer splices the output of the second layer at time steps  $(t-3)$  to  $t$  and  $t$  to  $(t+3)$ . The fourth and the fifth layer keep the same temporal context. Therefore, the total temporal context after the third layer is 15 frames [12].

The statistical pooling layer converts the variable-length input into a fixed-dimensional vector. It aggregates over the output vectors from the fifth frame layer and computes their mean and standard deviation [12].

The segment layers are hidden layers that map the output of the statistical pooling layer to speaker identities. Finally, the softmax layer returns the probability distribution across speakers available in the training set [12].

In the enrolment phase, the x-vector system is trained to solve a classification problem. A speaker segment is input to the model. The model outputs a probability distribution across all the known speaker labels. To use this model as an embedding extractor, the last layer is discarded together with the non-linearity (ReLU) layer from the second segment layers, and the output is used as the speaker embedding for the speaker identification model [12].

Architecture of the x-vector system described by Snyder et al. is depicted in Figure 9.

## 2.4.3 ECAPA-TDNN

ECAPA-TDNN is a version of the previously described x-vector system with enhancements adapted from recent trends in face verification and computer vision. The network

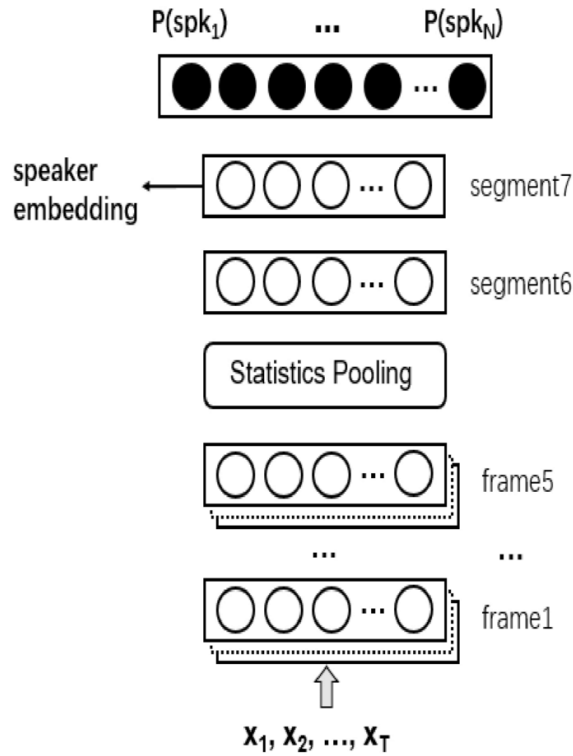


Figure 9. Architecture of the x-vector system [13].

architecture is depicted in Figure 10. The ECAPA-TDNN architecture has two main differences:

- Channel- and context-dependent statistics pooling: the statistical pooling layer is replaced with an attention mechanism and adapted to be channel-dependent. Success with multi-headed attention has shown that certain speaker properties can be extracted on different sets of frames. The attentive statistical pooling allows the network to focus only on frames it deems important. Furthermore, making the attention mechanism channel-dependent allows focusing only on speaker characteristics that do not activate on identical or similar time instances [14].
- 1-dimensional squeeze-excitation Res2Blocks: since the temporal context of the initial x-vector system is limited to 15 frames, and it is proven to be beneficial to expand the temporal context, the frame-level layers are replaced with a Conv1D + ReLU + Batch normalization block, three SE-Res2Blocks with residual connections, following another Conv1D + ReLU block before the attentive statistical pooling layer [14].

### Conv1D

Conv1D layer stands for 1-dimensional convolutional layer, also known as temporal convolution. It is a set of 1-dimensional filters (or kernels), parameters of which are

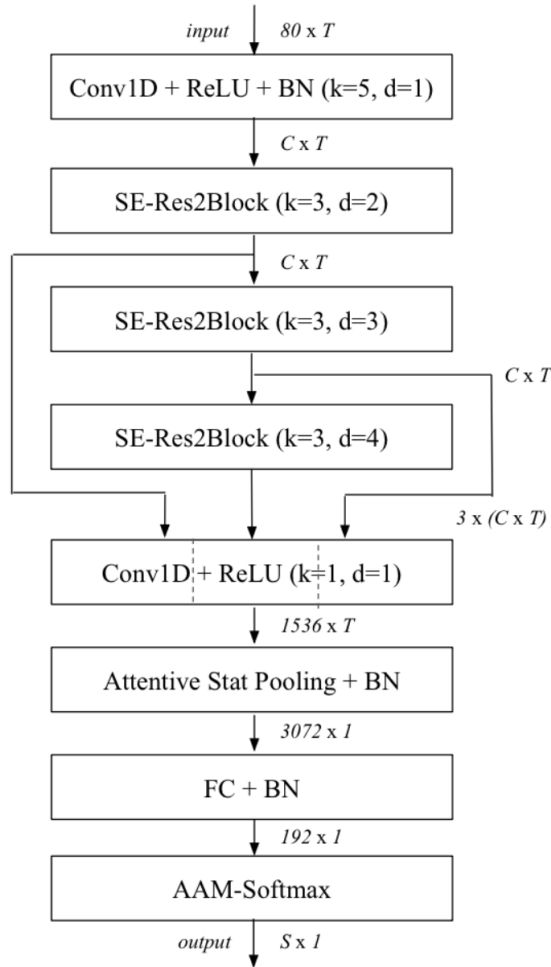


Figure 10. Network topology of the ECAPA-TDNN [14].

learned throughout the training. It convolves over a single dimension in the input and produces a transformed output depending on the learned weights.

## ReLU

ReLU stands for rectified linear unit. ReLU is a simple activation function commonly used in neural networks that helps to model non-linearities in the data.

## Batch Normalization

Batch normalization simply centres and re-scales the layer inputs to make the neural network training process more stable and faster [15]. It also helps to avoid the exploding and/or vanishing gradients problem in neural networks.

## **Squeeze and Excite (SE)**

Convolutional layers allow learning multiple feature maps from a single input (channel) as the layers go deeper by increasing the number of channels in the output of a convolutional layer. However, traditionally these channels are all weighted equally. Squeeze-and-Excitation technique allows to selectively emphasize channels with more informative features [16].

### **SE-Res2Block**

SE-Res2Block consists of four blocks [16]:

- Conv1D + ReLU + Batch normalization
- Res2 Dilated Conv1d + ReLU + Batch normalization: Dilation adds a configurable spacing between each element in the input
- Conv1D + ReLU + Batch normalization
- SE block

## **3. Data**

This chapter describes the data used in the speaker identification experiments. The data consists of audio recordings and metadata associated with each recording from two publicly available sources: Estonian Public Broadcasting archive and Soundcloud.

### **3.1 Sources**

#### **3.1.1 Estonian Public Broadcasting Archive**

Estonian Public Broadcasting archive is a collection of radio and TV shows that have been produced and broadcasted by Estonian Public Broadcasting (ERR) over the years. Access to the archive is free of charge. Each radio episode contains its metadata, including the title, description and names of the speakers in the episode. TV shows do not include any metadata besides the title, therefore the names of the speakers had to be labelled manually.

The thesis uses data from three different radio shows: "Päevakaja", "Reporteritund", and "Uudised". Additionally, six episodes from both "Aktuaalne Kaamera" and "Ringvaade" are used.

#### **3.1.2 Soundcloud**

Soundcloud is an audio distribution platform, where users can share and listen to music, podcasts, and other audio content from independent creators.

To evaluate model performance on out-of-domain datasets, a set of episodes are used from the "Arvamusfestival" channel. "Arvamusfestival" is an annual open-air event, where people can discuss and debate a range of topics and issues that are important to society, including politics, culture, environment, education, and more. These debates are recorded and published through the Soundcloud platform.

Each episode includes only description for each episode, which may or may not include the names of the speakers in the episode.



## 3.2 Data Acquisition

Two scrapers were built in order to obtain the data from the aforementioned sources.

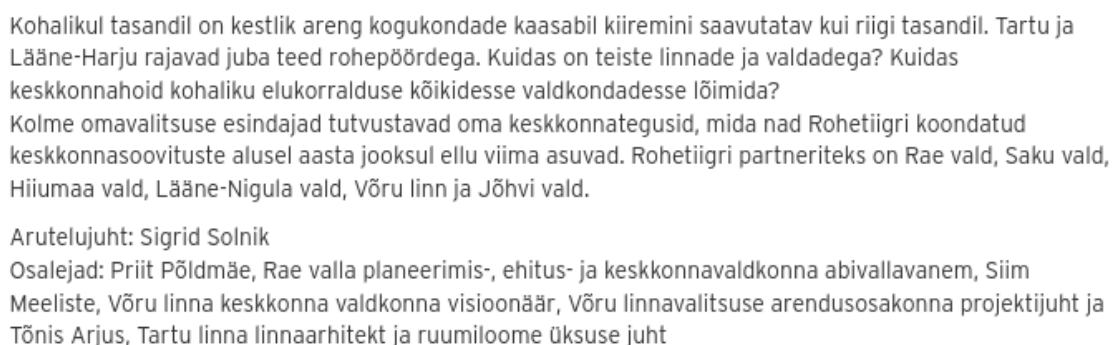
Firstly, the ERR archive scraper takes a list of URLs referencing the radio shows in interest. The scraper iterates over all the shows and episodes, downloads the audio files and extracts metadata visible on the page. Names of the speakers were formatted to a unified format, i.e. *{first name} {last name}*.

Similar scraper was used for TV shows, however none of the TV shows had any metadata associated. Six episodes from "Aktuaalne Kaamera" and six episodes from "Ringvaade" were randomly selected and manually labelled with the names of the speakers.

Secondly, the Soundcloud scraper takes a reference to a Soundcloud profile and downloads all the tracks with corresponding descriptions uploaded by that user.

Since the system is trained to identify Estonian Public Figures and most of the content is in Estonian, a language identification model was used to exclude episodes of the "Arvamusfestival" in foreign languages. The language identification model is a version of RoBERTa, called XLM-RoBERTa, which is trained on 2.5TB of CommonCrawl data containing 100 languages using masked language modelling (MLM) objective.

On the purged dataset, a named entity recognition model was used to recognize names of the speakers from the episode descriptions. Since the names of the speakers were arbitrarily located in the description (see Figure 11), a rule-based approach would not have been effective. The model used for named entity recognition was XLM-R + NER (another version of XLM-RoBERTa), which is fine-tuned for named entity recognition on XTREME dataset containing data across 40 languages.



Kohalikul tasandil on kestlik areng kogukondade kaasabil kiiremini saavutatav kui riigi tasandil. Tartu ja Lääne-Harju rajavad juba teed rohepöördega. Kuidas on teiste linnade ja valdadega? Kuidas keskkonnahoid kohaliku elukorralduse kõikidesse valdkondadesse lõimida? Kolme omavalitsuse esindajad tutvustavad oma keskkonnategusid, mida nad Rohetiigri koostatud keskkonnasoovituste alusel aasta jooksul ellu viima asuvad. Rohetiigri partneriteks on Rae vald, Saku vald, Hiiumaa vald, Lääne-Nigula vald, Võru linn ja Jõhvi vald.

Arutelujuht: Sigrid Solnik  
Osalejad: Priit Põldmäe, Rae valla planeerimis-, ehitus- ja keskkonnavaldkonna abivallavanem, Siim Meeliste, Võru linna keskkonna valdkonna visioonäär, Võru linnavalitsuse arendusosakonna projektijuht ja Tõnis Arjus, Tartu linna linnaarhitekt ja ruumiloome üksuse juht

Figure 11. Description of an episode in "Arvamusfestival".

Unfortunately, some speakers who appeared in the recordings were not mentioned in the descriptions and caused false-positives in the model validation phase. Thus, ten episodes with the largest speaker sets were picked and manually verified from the "Arvamusfestival" dataset.

### 3.3 Statistics

#### 3.3.1 Total number of shows used

The total number of in-domain recordings is close to 24 000. The put-of-domain recordings include 32 recordings from ERR TV shows and "Arvamusfestival", which were hand-picked and whose labels were manually verified after an automatic extraction. Datasets were split into train, dev, and test sets. The train set is used for training the model. The dev set is used for model training progress monitoring and validation. The test set is used for the trained model evaluation. Table 1 displays the total number of recordings used in each set.

Show	Train	Dev	Test	Total
Uudised	10585	1323	1324	13232
Päevakaja	7109	889	889	8887
Reporteritund	1236	154	155	1545
TV (Aktuaalne Kaamera, Ringvaade)	0	6	6	12
Arvamusfestival	0	0	20	20

Table 1. Total number of recordings per show and dataset.

#### 3.3.2 Total duration of recordings

In total almost 4490 hours of recordings was acquired. The distribution of the recording durations is shown in Table 2.

Show	Train	Dev	Test	Total
Uudised	27492	3110	3207	33809
Päevakaja	118655	18945	19537	157137
Reporteritund	59973	8049	8187	76209
TV (Aktuaalne Kaamera, Ringvaade)	0	173	171	344
Arvamusfestival	0	0	1861	1861

Table 2. Total duration of recordings per show and dataset in minutes.

### 3.3.3 Number of recordings annually

Figure 12 shows the number of recordings annually for each show in the dataset. "Reporteritund" dates back to year 1957, but the number of shows recorded varies annually. "Uudised" is a show with the highest number of recordings. The published recording frequency has increased almost 10 times since the last few years.

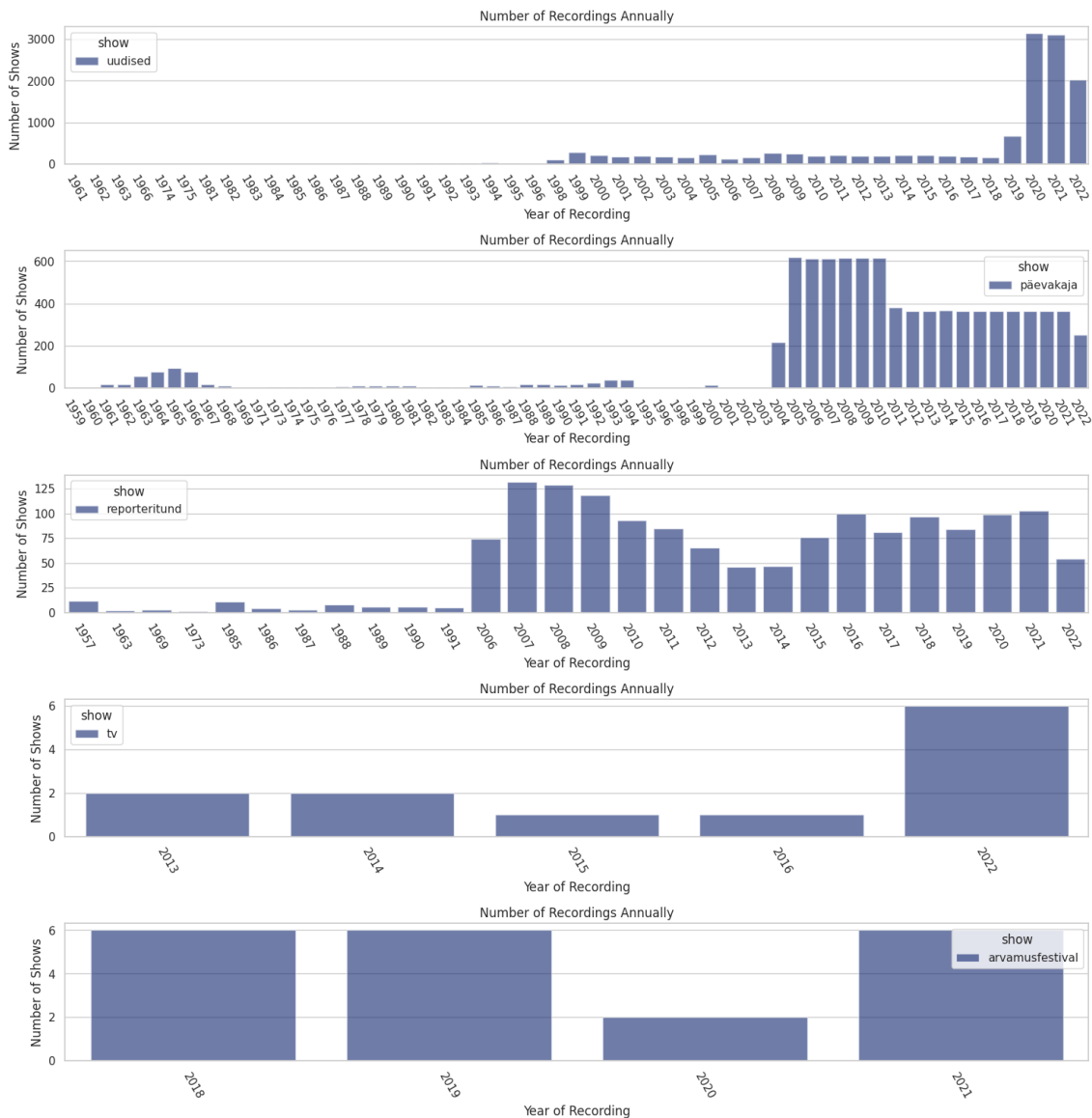


Figure 12. Number of recordings per show annually.

### 3.3.4 Average number of speaker occurrences per show annually

Figure 13 shows how the average number of speaker occurrences in a recording has changed over the years per show. While most of the shows include a similar number of

speakers on average in every episode, the number of speaker occurrences per episode has increased in "Päevakaja" since 2009.

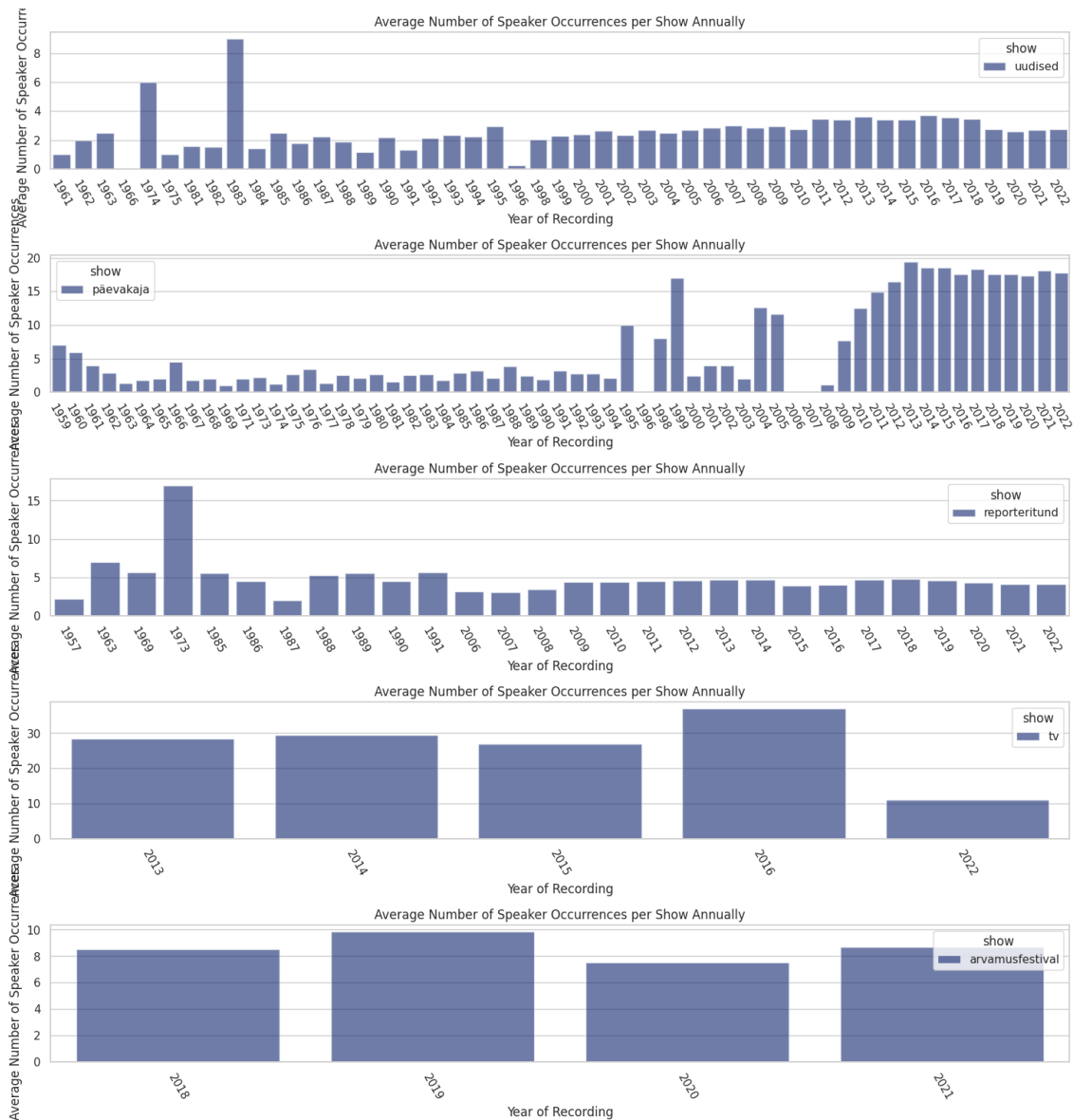


Figure 13. Average number of speaker occurrences in a recording per show annually.

### 3.3.5 Frequency Rank vs. Appearances

Figure 14 depicts how the speaker occurrence frequency rank compares to the number of occurrences across all episodes in a show. For example, it is shown that in "Päevakaja" the most frequent speaker appears in around 400 episodes more compared to the second most frequent speaker.

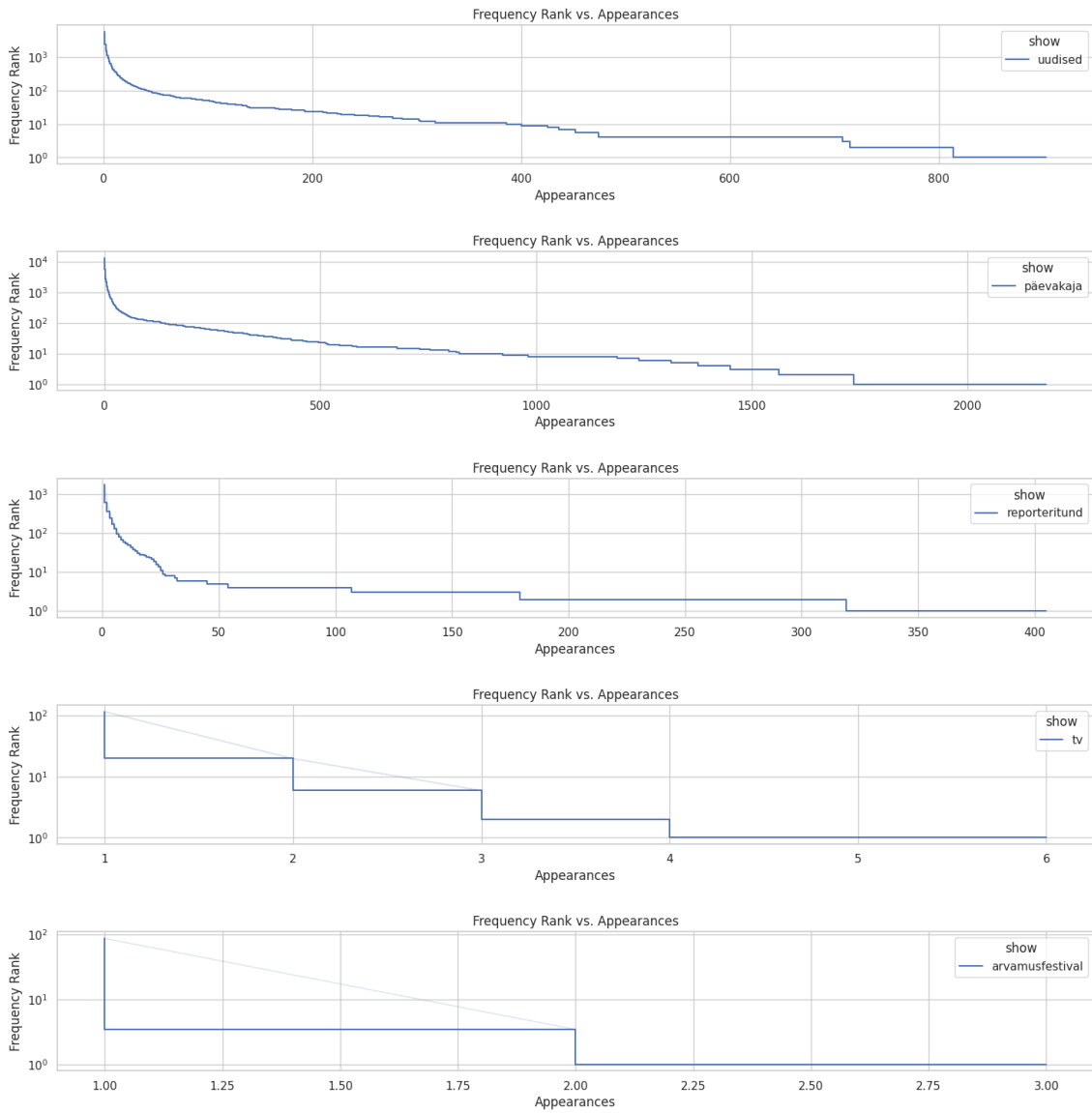


Figure 14. Average number of speaker occurrences per show.

### 3.3.6 Most Frequent Speakers per Show

Figures 3-7 display top ten most frequent speakers in the show. It is shown that out-of-domain shows ("Arvamusfestival" and TV) do not include any frequent speakers that appear in in-domain shows ("Uudised", "Päevakaja", and "Reporteritund").

Show	Name	Count	Rank
Uudised	Uku Toom	903	1
Uudised	Madis Hindre	814	2
Uudised	Mall Mälberg	715	3
Uudised	Indrek Kiisler	708	4
Uudised	Margitta Otsmaa	474	5
Uudised	Ester Vilgats	474	5
Uudised	Jüri Nikolajev	452	7
Uudised	Joakim Klementi	436	8
Uudised	Kai Vare	425	9
Uudised	Olev Kenk	400	10

Table 3. Top 10 most frequent speakers in "Uudised".

Show	Name	Count	Rank
Päevakaja	Uku Toom	2182	1
Päevakaja	Mall Mälberg	1736	2
Päevakaja	Tõnu Karjatse	1562	3
Päevakaja	Margitta Otsmaa	1450	4
Päevakaja	Kai Vare	1376	5
Päevakaja	Indrek Kiisler	1312	6
Päevakaja	Riina Eentalu	1238	7
Päevakaja	Janek Salme	1188	8
Päevakaja	Madis Hindre	981	9
Päevakaja	Olev Kenk	922	10

Table 4. Top 10 most frequent speakers in "Päevakaja".

Show	Name	Count	Rank
Reporteritund	Arp Müller	405	1
Reporteritund	Kaja Kärner	319	2
Reporteritund	Mirko Ojakivi	179	3
Reporteritund	Lauri Hussar	107	4
Reporteritund	Andrus Ansip	54	5
Reporteritund	Peeter Kaldre	45	6
Reporteritund	Neeme Raud	32	7
Reporteritund	Urmas Paet	31	8
Reporteritund	Mart Ummelas	27	9
Reporteritund	Harri Tiido	26	11

Table 5. Top 10 most frequent speakers in "Reporteritund".

Show	Name	Count	Rank
TV	Marko Reikop	6	1
TV	Anna Pihl	4	2
TV	Martin Mileiko	3	6
TV	Maria-Ann Rohemäe	3	6
TV	Margus Saar	3	6
TV	Priit Kuusk	3	6
TV	Kadri Hinrikus	3	6
TV	Taavi Rõivas	3	6
TV	Tiina Jaakson	3	6
TV	Astrid Kannel	2	20

Table 6. Top 10 most frequent speakers in TV.

Show	Name	Count	Rank
Arvamusfestival	Mailis Reps	3	1
Arvamusfestival	Kristi Ockba	2	3.5
Arvamusfestival	Züleyxa Izmailova	2	3.5
Arvamusfestival	Urmast Viilma	2	3.5
Arvamusfestival	Märt Treier	2	3.5
Arvamusfestival	Aaro Nursi	1	88.5
Arvamusfestival	Kaisa Jõgeva	1	88.5
Arvamusfestival	Kaupo Heinma	1	88.5
Arvamusfestival	Katrin Helendi	1	88.5
Arvamusfestival	Katri Lamesoo	1	88.5

Table 7. Top 10 most frequent speakers in "Arvamusfestival".

## 4. Experimental Setup

This chapter describes the experimental setup for the weakly supervised speaker identification system.

### 4.1 Overview

As a baseline a deep neural network is implemented that expects speaker embeddings as an input, solves a multi-label classification problem, and outputs the probability distribution across the speakers known from the training set. The speaker label with the highest probability is chosen as the identity for the input speaker embedding.

The baseline model uses static i-vector embeddings extracted from audio files using the Kaldi toolkit. The extraction process is explained more in detail in Chapter 5.

The speaker identification model is written using the PyTorch deep learning library. The data structure and pre-processing is managed using the Kaldi toolkit.

### 4.2 Model Architecture

The model architecture is a simple fully-connected deep neural network and consists of the following layers:

- Linear: Applies linear transformation on the input data  
Input: (batch size, embedding dimensionality).  
Embedding dimensionality depends on the type of embeddings used (2048 for i-vectors and 192 for x-vectors).
- Leaky ReLU: Activation function that allows to learn non-linearities in the input data. Similar to ReLU, but has a small slope for negative values instead of zero.
- Dropout: Disables a random subset of neurons on every training step to avoid overfitting. By default, 10 % of the neurons are randomly disabled.
- Linear:  
Input: (batch size, hidden dim)  
Hidden dim is a configurable number of dimensions defined as a hyperparameter before training.
- Leaky ReLU



- Dropout
- Linear:
  - Input: (batch size, hidden dim)
  - Output: (batch size, number of speakers in training set).
- Soft-max: Normalizes the output probability distribution, so the values sum to 1.

The model is implemented using the PyTorch Lightning framework. Lightning abstracts away most of the boilerplate code that is typically required for training a deep neural network. At the same time it allows to easily override every step in the training process.

### 4.3 KaldiDataset

The dataset loader was implemented using the PyTorch Dataset abstraction, which defines functions how to load the data into memory from disk in batches. Batching allows to train and inference the model on datasets that would not normally fit into operational memory or video ram.

Since the dataset structure follows the standard Kaldi format, a generic KaldiDataset class was implemented. KaldiDataset requires *wav2names.json* and *wav2spk* files to be present in the provided dataset path.

KaldiDataset is responsible for the following tasks:

- Mapping speaker names to sequential label identifiers.
- Counting the number of speaker occurrences in the dataset.
- Computing the oracle name coverage: In the training phase speakers that occur less than a configurable *min-speaker-occ* times are excluded from the dataset. Oracle name coverage is the ratio of names left in the dataset. This is also the theoretical maximum recall the model can achieve during the evaluation phase.
- Loading speaker embeddings from the disk.

### 4.4 Training

The training process is similar to the weakly supervised speaker identification method described in Chapter 1. A batch of embeddings is propagated through the aforementioned network. The model calculates label regularized loss during training based on the previous work of Martin Karu and Tanel Alumäe. Stochastic gradient descent optimizer minimizes the loss function through back-propagation. This process is repeated over each batch and

for a fixed number of times (epochs). [1]

## **4.5 Inference**

Once the model is trained, a new speaker embedding can be extracted from a new speech segment, which can be input to the network to calculate the probability distribution across the known speaker identities.

In order to select a single speaker identity from the probability distribution as a prediction, a fixed threshold is used that can be tuned after the model is trained. The output values always sum to one. Threshold allows to pick a single label based on the probability distribution if the label has higher probability than the threshold compared to other labels in the output. Otherwise, the predicted label is unknown.

During the evaluation phase, precision and recall is measured at a fixed 50 % threshold. Precision measures how many speakers the model predicted correctly from all of its predictions. Recall measures how many speakers the model could recognize at all. In this thesis, the goal is to optimize for high precision while also trying to increase recall.

## 5. Improvements

This section proposes several improvements to the baseline model described in Chapter 4. The i-vector-based model is re-trained on the dataset described in Chapter 3. The posterior probabilities are adjusted, and data augmentation is applied to improve the model's performance. Furthermore, i-vector embedding extractor is replaced with x-vectors embedding extractor using a pre-trained ECAPA-TDNN model. Moreover, the pre-trained ECAPA-TDNN model is fine-tuned to fit the data collected in this thesis. Finally, dynamic audio-level augmentation is performed using spec-augmentation. [17]

### 5.1 Adjusting the Posterior Probabilities

Adjusting the posterior probabilities is a regularization technique used to improve generalization performance in machine learning methods when the training data is highly imbalanced. The general idea is to penalize the model's output probability distribution depending on the number of occurrences of the speakers in the training data. Machine learning methods can bias towards the more frequent speakers. That means during inference, the model is overly confident over the frequent speakers, but not very confident about the speakers seen only a few times in the training set.

Let's obtain the true prior probabilities for each class,  $P'(y = k)$ , and the imbalanced prior probabilities,  $P(y = k)$ , where  $k$  is the speaker label.

For each class  $k$ , the ratio of true priors to imbalanced priors can be computed:

$$R_k = \frac{P'(y=k)}{P(y=k)}$$

The given the output probabilities of the speaker identification model,  $P(y = k|x)$ , can be adjusted by using the following formula:

$$P'(y = k|x) = \frac{P(y=k|x) \cdot R_k}{Z}$$

where  $Z$  is the normalization constant to ensure the sum of the adjusted probabilities equals 1:

$$Z = \sum_k P(y = k|x) \cdot R_k$$

## 5.2 Data Augmentation

The performance of deep neural network-based models relies heavily on the amount of data used for training. Growing the size of the labelled dataset can be expensive, therefore data augmentation is often used to prevent overfitting and to improve the generalization ability [18].

In this thesis, each recording was modified in the training set by adding following noises to the audio files, each with 80 % probability:

- Reverberation (Room Impulse Response, Echo): A repeated vanishing reflection of sound after it is produced [19].
- Background noise: A random Gaussian noise - makes input space smoother and easier to learn.
- Point-source noise: Random sudden noises (e.g. door slams and footsteps).

The modified recordings were combined with the original recordings and used for training with two times larger number of epochs.

## 5.3 Pre-Trained Models

Due to the lack of available in-domain data, training a large deep neural network from scratch that performs well is nearly impossible. In these cases transfer learning is commonly used to adapt pre-trained models to use-cases with smaller datasets. Transfer learning consists of two steps: training and fine-tuning. The model is usually trained to solve a general task on a very large dataset that is easy to acquire. These pre-trained models are also often made publicly available for research purposes. Finally, the pre-trained model is fine-tuned to a specific use-case on a smaller dataset with a smaller learning rate.

The implementation in this thesis uses a pre-trained a TDNN-UBM i-vector extractor and a pre-trained ECAPA-TDNN x-vector extractor.

### 5.3.1 Kaldi's TDNN-UBM I-vectors

The baseline model uses i-vector extractor provided by the Kaldi toolkit. Kaldi is toolkit that provides scripts and libraries for common speech recognition tasks. Kaldi's i-vector extractor uses TDNN-UBM (Time delay deep neural network-based universal background model). TDNN-UBM is a method similar to GMM-UBM described in Chapter 2, however

the speakers are modelled using a time delay neural network instead of a Gaussian mixture model [20].

Kaldi toolkit has strict requirements how the dataset should be stored in order to use its scripts. Fortunately, it also provides scripts to transform the raw audio and metadata files to the standard structure expected by Kaldi. The data directory is split by show and includes components including:

- `/wav/`: Directory containing the audio files in waveform audio file format (wav).
- `/wav.scp`: File that defines a mapping between an audio file identifier and the wav file location.
- `/wav2names.json`: File that defines which speakers speak in each audio file.

To extract i-vector embeddings from raw audio files, Kaldi provides a script called `extract_ivectors.sh`. The script outputs a `spk_ivector.scp` file, which contains the extracted embeddings for every diarized speaker. These embeddings can be used to train and inference the weakly supervised speaker identification model.

### 5.3.2 Speechbrain’s ECAPA-TDNN X-vectors

ECAPA-TDNN was proposed by Brecht Desplanques et al. in 2020 and showed 19 % relative improvement in equal-error-rate compared to the strong baseline systems in VoxCeleb and VoxSRC 2019 evaluation sets. This thesis uses x-vectors from the SpeechBrain’s ECAPA-TDNN model [21] pre-trained on the Voxceleb dataset [22].

To acquire the pre-trained model, *speechbrain* library provides *EncoderClassifier* class, which was used to download the pre-trained *speechbrain/spkrec-ecapa-voxceleb* model. To store the x-vector embeddings in the same format as i-vector embeddings, the *kaldiio* library provides *WriteHelper* class that allows to store the embeddings in the format expected by Kaldi. The stored embeddings were used in the same manner to train the weakly supervised speaker identification model.

## 5.4 Fine-Tuning ECAPA-TDNN

Instead of training the speaker identification model separately from the embedding extractor model, the ECAPA-TDNN model is used as a backbone for the speaker identification model. This way the embedding extractor can be fine-tuned while training the speaker identification model.

In order to combine these two models, the KaldiDataset was modified by adding the ECAPA-TDNN pre-trained model as a backbone to the weakly supervised speaker identification model.

KaldiDataset now loads the raw audio files in addition to their metadata. During training, two five-second samples were randomly picked from each utterance on every epoch.

The speaker identification model now downloads the pre-trained ECAPA-TDNN model. On forward propagation, the model is input the five-second audio segments instead of the static speaker embeddings. The model first calculates features (MFCCs) from the raw audio file, applies mean variance normalization, and computes embeddings using the ECAPA-TDNN network. The rest of the model architecture remains the same.

Fine-tuning a pre-trained model should not be done using as high learning rate as the rest of the model. The learning rate is scaled down to 1 % only for the backbone model. The learning rate is also frozen after 3000 training steps to avoid overfitting the pre-trained model.

## 5.5 SpecAugment

Now that the speaker identification model is trained together with the x-vectors extractor, dynamic data augmentation can be applied in the training phase as well. SpecAugment is one of the most widely used audio augmentation methods today. SpecAugment (see Figure 15) applies the following transformations on the extracted features (MFCCs) spectrogram:

- Time stretch: Scaling the input along the time domain.
- Time masking: Masks  $n$  time windows across the whole frequency domain.
- Frequency masking: Masks  $n$  frequency bands across the whole time domain.

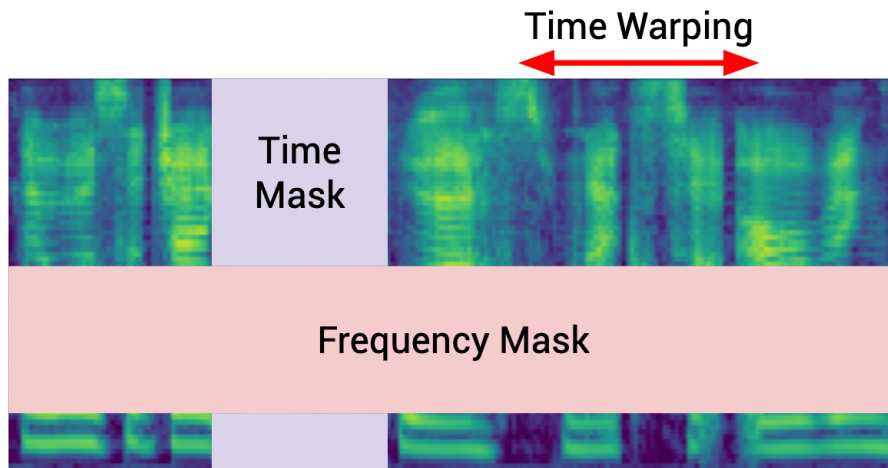


Figure 15. Spec-Augment applied on an audio spectrogram [23].

This thesis uses only time masking and frequency masking, because the model is trained strictly on five-second audio samples. Masking is applied after mean variance normalization for two frequency bands and two time windows.

After applying the dynamic spec-augmentation, the model was trained two times longer than normally, so the model would see more than one combination of the augmented data.

## 6. Evaluation

This chapter explains how each of the aforementioned improvements affected the baseline model performance. Precision and recall is used to evaluate each model improvement. All predictions use a fixed 50 % threshold, i.e. if the model is 50 % more confident about a label compared to the other labels, it is chosen as the prediction. Otherwise, the predicted speaker identity is unknown.

The goal is to optimize the speaker identification system for precision in order to avoid false-positives as much as possible. The precision can be further increased by increasing the threshold, however this increase will sacrifice recall, i.e. fewer speakers will be recognized.

### 6.1 Baseline

As a baseline, the weakly supervised speaker identification model proposed by Martin Karu and Tanel Alumäe is reimplemented in PyTorch [1]. The model was previously trained only on a "Päevakaja" dataset acquired from ERR archive in 2017. Table 8 shows the performance metrics per show on the newly acquired datasets. It is shown that precision on "Päevakaja" is still higher than 90 %, however it produces a lot of false-positives for out-of-domain datasets.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	77.8 %	45.4 %	69.8 %	41.1 %
Päevakaja	95.2 %	59.6 %	90.5 %	48.1 %
Reporteritund	75.6 %	58.8 %	74.2 %	56.4 %
TV	63.7 %	27.3 %	69.0 %	23.4 %
Arvamusfestival			43.1 %	17.2 %

Table 8. Baseline model performance metrics.

### 6.2 Larger and More Recent Dataset

It is expected that people in the Estonian Public Broadcasting shows change over time. That reduces the recall of the model over time. New speakers with similar voices may also be confused with known speakers, which makes the model produce more false-positives. After re-training the model on the larger and more recent datasets, the precision and recall increased for all shows, as seen in Table 9.



The lowest precision for in-domain data increased to 93.8 % and for out-of-domain data increased to 87 %. It is also shown that recall for out-of-domain shows is much lower compared to the in-domain shows. That is expected since not all speakers are present in the in-domain datasets.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	98.7 %	52.6 %	96.4 %	45.4 %
Päevakaja	97.2 %	58.8 %	98.7 %	62.0 %
Reporteritund	93.8 %	54.8 %	94.1 %	57.2 %
TV	89.3 %	18.5 %	87.0 %	17.9 %
Arvamusfestival			95.7 %	12.4 %

Table 9. Re-trained model performance metrics on a larger dataset.

### 6.3 Adjusting the Posterior Probabilities

Next, the output probabilities for more frequent speaker labels were penalized to remove bias towards the more frequent speakers in the training set. Table 10 shows the re-trained model performance metrics with adjusted posterior probabilities. It is shown that the adjustment increases precision, but lowers recall for all datasets. The lowest precision for in-domain datasets increased to 96.1 % and for out-of-domain datasets increased to 92.3 %. The precision now is in the expected region, however there is still improvement for recall.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	99.6 %	43.8 %	99.1 %	37.1 %
Päevakaja	97.8 %	50.7 %	99.3 %	55.5 %
Reporteritund	97.6 %	47.2 %	96.1 %	46.7 %
TV	93.3 %	10.4 %	92.3 %	10.7 %
Arvamusfestival			100 %	10.2 %

Table 10. Re-trained model performance metrics with adjusted posterior probabilities.

### 6.4 Speechbrain’s ECAPA-TDNN

Next, Kaldi’s i-vector embeddings were replaced by Speechbrain’s ECAPA-TDNN’s x-vector embeddings. Table 11 shows the performance metrics of the speaker identification model based on x-vectors. Table 12 uses also adjusted posterior probabilities.

Table 11 shows the highest recall so far without sacrificing a lot on precision. It is shown in Table 12 that posterior probability adjustment helps to increase precision for out-of-domain

datasets, but at the same time lowers recall too much, therefore weighting the adjustment should be considered.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	98.4 %	56.4 %	96.3 %	49.9 %
Päevakaja	97.6 %	62.0 %	98.8 %	63.7 %
Reporteritund	93.2 %	57.7 %	94.2 %	59.1 %
TV	90.3 %	20.7 %	89.7 %	23.2 %
Arvamusfestival			92.1 %	19.8 %

Table 11. X-vector-based model performance metrics.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	99.0 %	40.9 %	98.9 %	36.3 %
Päevakaja	98.0 %	44.4 %	99.3 %	48.8 %
Reporteritund	98.3 %	41.7 %	96.3 %	41.2 %
TV	100 %	11.1 %	100 %	11.6 %
Arvamusfestival			100 %	12.4 %

Table 12. X-vector-based model performance metrics with posterior probability adjustment.

## 6.5 Data Augmentation

Next, the model was trained on the augmented dataset. Tables 13 and 14 show the model performance metrics trained on the embeddings from the augmented dataset. Data augmentation helped to increase precision and recall for in-domain datasets, but not enough for out-of-domain datasets.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	98.9 %	56.9 %	97.6 %	50.9 %
Päevakaja	97.8 %	62.5 %	98.9 %	63.7 %
Reporteritund	95.4 %	58.0 %	93.7 %	58.0 %
TV	87.1 %	20.0 %	87.1 %	24.1 %
Arvamusfestival			86.5 %	18.1 %

Table 13. Model performance metrics with data augmentation.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	99.0 %	41.7 %	99.0 %	36.7 %
Päevakaja	97.8 %	44.2 %	98.9 %	48.5 %
Reporteritund	97.9 %	34.3 %	94.2 %	33.2 %
TV	100 %	16.3 %	92.9 %	11.6 %
Arvamusfestival			100 %	12.4 %

Table 14. Model performance metrics with data augmentation and posterior probability adjustment.

## 6.6 Fine-Tuned ECAPA-TDNN

Next, the static embeddings were replaced by fine-tuning the x-vector extractor while training the speaker identification model. Tables 15 and 16 show the model performance metrics with fine-tuned x-vector extractor. Fine-tuning the embedding extractor increased precision and recall for all datasets.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	98.7 %	56.9 %	97.9 %	51.1 %
Päevakaja	98.1 %	63.2 %	98.9 %	64.8 %
Reporteritund	95.0 %	58.0 %	94.3 %	59.7 %
TV	93.8 %	22.2 %	93.5 %	25.9 %
Arvamusfestival			97.4 %	21.5 %

Table 15. Model performance metrics with fine-tuning.

Show	dev		test	
	Precision	Recall	Precision	Recall
Uudised	98.2 %	42.5 %	98.8 %	35.7 %
Päevakaja	97.2 %	46.3 %	98.5 %	50.2 %
Reporteritund	98.1 %	44.6 %	95.4 %	45.3 %
TV	93.3 %	10.4 %	92.9 %	11.6 %
Arvamusfestival			100 %	17.5 %

Table 16. Model performance metrics with fine-tuning and posterior probability adjustment.

## 6.7 SpecAugment

Finally, spec-augmentation was applied to the x-vector extractor and the model was trained for two times higher number of epochs. Tables 17 and 18 show the model performance

metrics with spec-augmented embedding extractor. This change achieved the highest recalls so far with acceptable precisions across all datasets. For out-of-domain datasets it is over-confident, however the precision can be increased by increasing model threshold.

	dev		test	
Show	Precision	Recall	Precision	Recall
Uudised	98.4 %	57.7 %	97.8 %	52.0 %
Päevakaja	98.1 %	63.7 %	99.0 %	65.3 %
Reporteritund	95.8 %	59.3 %	94.6 %	60.7 %
TV	92.5 %	27.4 %	91.2 %	27.7 %
Arvamusfestival			95.2 %	22.6 %

Table 17. Model performance metrics with SpecAugment.

	dev		test	
Show	Precision	Recall	Precision	Recall
Uudised	98.9 %	43.3 %	99.2 %	38.1 %
Päevakaja	97.5 %	47.8 %	98.8 %	51.2 %
Reporteritund	97.0 %	47.8 %	95.3 %	47.3 %
TV	91.7 %	16.3 %	94.7 %	16.1 %
Arvamusfestival			97.1 %	18.6 %

Table 18. Model performance metrics with SpecAugment and with posterior probability adjustment.

## 6.8 Threshold Tuning

Based on the precisions and recalls above, F1 scores were calculated across all model versions and datasets. The SpecAugmented ECAPA-TDNN model produced the highest relative improvement across all datasets. The relative improvements are visible in Table 19.

	Uudised		Päevakaja		Reporteritund		TV		Arvamusf.
	dev	test	dev	test	dev	test	dev	test	test
Model	F1 relative improvement								
i-vector	19.7%	19.2%	0.0%	21.3%	4.6%	11.0%	-19.7%	-15.2%	-10.3%
ecapa	25.0%	27.1%	3.4%	23.3%	7.8%	13.4%	-11.7%	5.5%	32.7%
ecapa/aug	26.0%	29.3%	4.0%	23.3%	9.1%	11.8%	-14.9%	8.1%	21.9%
fine-tuned	25.9%	29.9%	4.8%	24.7%	8.9%	14.1%	-6.0%	16.1%	43.4%
spec-augm	26.8%	31.2%	5.4%	25.3%	10.8%	15.4%	10.6%	21.5%	48.9%

Table 19. Relative F1 measure baseline improvements across datasets.

The SpecAugmented ECAPA-TDNN model is used to tune the model threshold. Since F1

score is not interpretable enough, the thesis aims to maximize the recall at 95 % precision metric instead. This metric shows how many speakers the model can recognize in the dataset without falling below 95 % precision.

As seen in Figure 16, in order to reach at least 95 % precision across all the in-domain datasets, at minimum 60 % threshold should be used due to "Reporteritund" test set. It is also shown that the model struggles to reach 95 % precision for the TV dev set. However, setting 80 % threshold would be enough to reach the target precision for "Arvamusfestival" and TV test sets.

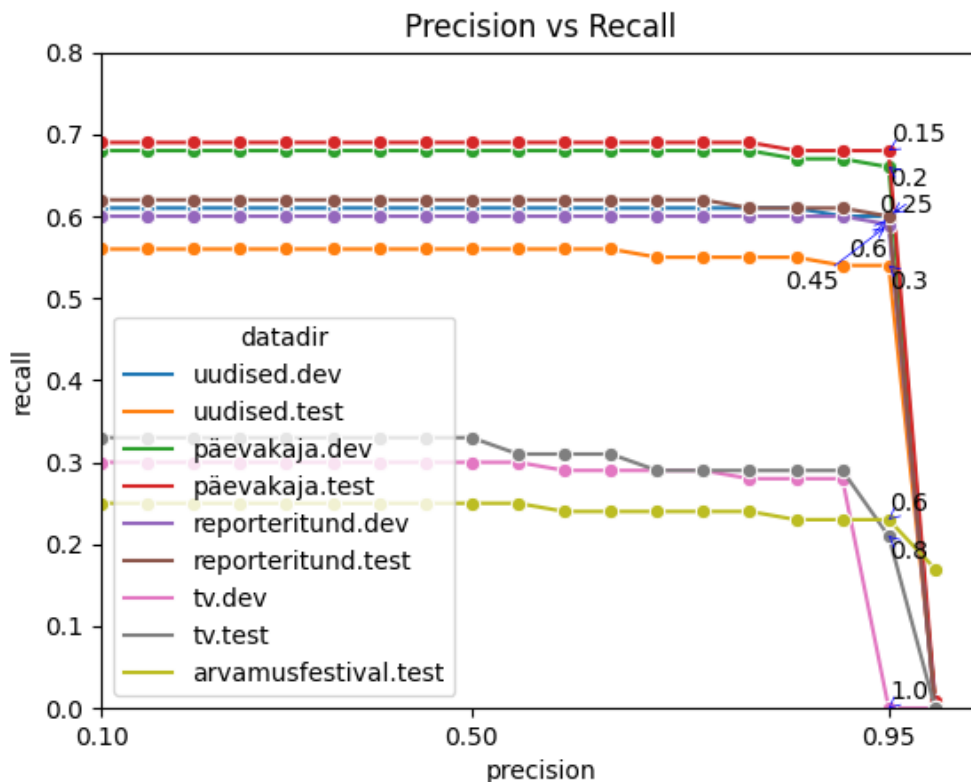


Figure 16. Precision vs. recall with thresholds at 95 % precision across datasets.

## 6.9 Summary

The main problem with the baseline model was low precision on all datasets except "Päevakaja". The final model was able to increase precision above 90 % on all datasets, even for shows the model has not seen during training. At the same time, the model was able to increase recall across all the datasets as well.

The baseline model was already tuned to achieve 95 % precision on an older set of

recordings from "Päevakaja". Comparing the baseline to the final spec-augmented ECAPA-TDNN model, it is shown in Table 19 that it achieved 5.4 % relative F1 improvement on "Päevakaja" dev set and 25.3 % on "Päevakaja" test set.

It is shown in Table 20 how much the SpecAugmented ECAPA-TDNN model improved both precision and recall across all datasets. The highest improvements in precision are seen for out-of-domain datasets, which was also the main objective in this thesis. Increased precision for out-of-domain dataset means that fewer speakers will be misidentified as the data evolves.

Show	Dataset	Metric	Improvement
Uudised	dev	Precision	26.4 %
Uudised	dev	Recall	27.0 %
Uudised	test	Precision	40.0 %
Uudised	test	Recall	26.5 %
Päevakaja	dev	Precision	3.0 %
Päevakaja	dev	Recall	6.9 %
Päevakaja	test	Precision	9.3 %
Päevakaja	test	Recall	35.8 %
Reporteritund	dev	Precision	26.7 %
Reporteritund	dev	Recall	0.9 %
Reporteritund	test	Precision	27.5 %
Reporteritund	test	Recall	7.6 %
TV	dev	Precision	45.2 %
TV	dev	Recall	0.4 %
TV	test	Precision	32.1 %
TV	test	Recall	18.3 %
Arvamusfestival	test	Precision	121.2 %
Arvamusfestival	test	Recall	31.8 %

Table 20. Relative precision and recall improvements across datasets.

## 7. Summary

The master thesis discussed background theory on speaker recognition and how speaker identification problem is solved today. It discussed the data acquisition process and provided statistics on the acquired data quality. Furthermore, it described the weakly-supervised speaker identification system used as a baseline, inspired from a previous master thesis by Martin Karu and Tanel Alumäe. It described several improvements to the baseline model, including data augmentation techniques, using and fine-tuning a state-of-the-art pre-trained x-vector embeddings extractor, and more. Finally, each improvement was evaluated to see how it affected the model performance.

For model performance evaluation, precision and recall were measured at a fixed 50 % precision. It was shown that the baseline model performed well on "Päevakaja" dataset, which it was trained on, but poorly on out-of-domain datasets. It was also shown that the baseline model produces more false-positives even for "Päevakaja" as new speakers appear in the recordings. Finally, the threshold was tuned on the best performing model to achieve at least 95 % precision on every in-domain dataset.

The hypothesis was that at least 90 % precision can be achieved on all datasets and at the same time recall can be increased on all datasets compared to the baseline model. In conclusion, the hypothesis is accepted based on the evaluation results seen in Chapter 6.

The main problem with the baseline model was low precision on out-of-domain audio recordings. That also means the baseline model produces more and more false-positives as new speakers appear in new recordings. Based on the out-of-domain dataset evaluations, the improved system allows maintaining higher precision compared to the baseline system as the data evolves and new speakers with similar voice characteristics appear.

The described baseline model is used today for automatic transcription system called "Kõnesalvestuste browser" (speech recordings browser), developed and maintained by TalTech Laboratory of Language Technology. The system collects new radio shows daily and transcribes them with names of Estonian public figures. This thesis provides practical output in terms of replacing the speaker identification system with the improved version from this thesis. Higher precision reduces the number of complaints about false-positives by the website visitors.

## References

- [1] Martin Karu and Tanel Alumäe. *Weakly Supervised Training of Speaker Identification Models*. 2018. arXiv: 1806.08621 [cs.LG].
- [2] Rafizah Mohd Hanifa, Khalid Isa, and Shamsul Mohamad. “A review on speaker recognition: Technology and challenges”. In: *Computers & Electrical Engineering* 90 (2021), p. 107005. ISSN: 0045-7906. DOI: <https://doi.org/10.1016/j.compeleceng.2021.107005>. URL: <https://www.sciencedirect.com/science/article/pii/S0045790621000318>.
- [3] Hao Tang, Stephen Chu, and Mark Hasegawa-Johnson. “Partially Supervised Speaker Clustering”. In: *IEEE transactions on pattern analysis and machine intelligence* 34 (Aug. 2011), pp. 959–71. DOI: 10.1109/TPAMI.2011.174.
- [4] Tumisho Mokgonyane et al. “Automatic Speaker Recognition System based on Machine Learning Algorithms”. In: Jan. 2019. DOI: 10.1109/RoboMech.2019.8704837.
- [5] Jia Min Karen Kua. “Improving automatic speaker verification using front-end and back-end diversity”. In: 2012.
- [6] Rizwan Rehman. “Auditory Scale Analysis and Evaluation of Phonemes in MISING Language”. In: *International Journal of Computer Applications* Vol. 113 Number 15 (Mar. 2015), pp. 1–5. DOI: 10.5120/19899-2001.
- [7] Donglai Zhu and K.K. Paliwal. “Product of power spectrum and group delay function for speech recognition”. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. 2004, pp. I–125. DOI: 10.1109/ICASSP.2004.1325938.
- [8] CHANRAN KIM. *MFCC Feature extraction for Sound Classification*. Kaggle. 2020. URL: <https://www.kaggle.com/code/seriousran/mfcc-feature-extraction-for-sound-classification>.
- [9] Najim Dehak et al. “Front-End Factor Analysis for Speaker Verification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011), pp. 788–798. DOI: 10.1109/TASL.2010.2064307.
- [10] Lei Lei and Kun She. “Identity Vector Extraction by Perceptual Wavelet Packet Entropy and Convolutional Neural Network for Voice Authentication”. In: *Entropy* 20 (Aug. 2018), p. 600. DOI: 10.3390/e20080600.



- [11] David Snyder et al. “X-Vectors: Robust DNN Embeddings for Speaker Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 5329–5333. DOI: 10.1109/ICASSP.2018.8461375.
- [12] Mickael Rouvier, Richard Dufour, and Pierre-Michel Bousquet. “Review of different robust x-vector extractors for speaker verification”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 1–5. DOI: 10.23919/Eusipco47968.2020.9287426.
- [13] Hao Wu et al. “A method of multi-models fusion for speaker recognition”. In: *International Journal of Speech Technology 25* (June 2022). DOI: 10.1007/s10772-022-09973-w.
- [14] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. “ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification”. In: Oct. 2020. DOI: 10.21437/Interspeech.2020-2650.
- [15] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167 [cs.LG].
- [16] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV].
- [17] Daniel S. Park et al. “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition”. In: *Interspeech 2019*. ISCA, Sept. 2019. DOI: 10.21437/interspeech.2019-2680. URL: <https://doi.org/10.21437%2Finterspeech.2019-2680>.
- [18] Shengyun Wei et al. “A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification”. In: *Journal of Physics: Conference Series 1453.1* (Jan. 2020), p. 012085. DOI: 10.1088/1742-6596/1453/1/012085. URL: <https://dx.doi.org/10.1088/1742-6596/1453/1/012085>.
- [19] Michael; Holly Hosford-Dunn; Ross J. Roeser Valente. *Audiology*. Thieme, 2008, pp. 425–426. ISBN: 978-1-58890-520-8.
- [20] David Snyder, Daniel Garcia-Romero, and Daniel Povey. “Time delay deep neural network-based universal background models for speaker recognition”. In: *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2015, pp. 92–97. DOI: 10.1109/ASRU.2015.7404779.
- [21] Mirco Ravanelli et al. *SpeechBrain: A General-Purpose Speech Toolkit*. arXiv:2106.04624. 2021. arXiv: 2106.04624 [eess.AS].

- [22] A. Nagrani, J. S. Chung, and A. Zisserman. “VoxCeleb: a large-scale speaker identification dataset”. In: *INTERSPEECH*. 2017.
- [23] Heng-Jui Chang. *Data Augmentation in Automatic Speech Recognition*. 2021. URL: <https://spectra.mathpix.com/article/2021.09.00002/asr-data-augmentation> (visited on 05/02/2023).

# Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis<sup>1</sup>

I Priit Käär

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Weakly Supervised Speaker Identification System Implementation based on Estonian Public Figures”, supervised by Tanel Alumäe
  - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
  - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

07.05.2023

---

<sup>1</sup>The non-exclusive licence is not valid during the validity of access restriction indicated in the student’s application for restriction on access to the graduation thesis that has been signed by the school’s dean, except in case of the university’s right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.