h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACULTY OF COMPUTER SCIENCE

Hochschule Darmstadt, University of Applied Science
– Biometrics and Internet-Security Research Group –

PRESENTATION ATTACK DETECTION FOR STATE-OF-THE-ART SPEAKER RECOGNITION SYSTEMS

Detection of Unit-Selection Attacks utilising Frequency Based Signal Analysis

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt von

ULRICH JOHANNES SCHERHAG

Referent:	Prof. Dr. Christoph Busch
Korreferent:	Andreas Nautsch
Ausgabedatum:	30.09.2015
Abgabedatum:	01.04.2016

Ulrich Johannes Scherhag: *Presentation Attack Detection for State-of-the-Art Speaker Recognition Systems,* Detection of Unit-Selection Attacks utilising Frequency Based Signal Analysis, Master of Science (M.Sc.),
© March 30, 2016

SUPERVISORS:
Prof. Dr. Christoph Busch
Andreas Nautsch

LOCATION:
Darmstadt

TIME FRAME:
30.09.2015 — 01.04.2016

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 30. März 2016

Ulrich Johannes Scherhag

ABSTRACT

In the modern society, biometrics is gaining more and more in importance, particularly facilitated by the increasing usage of multi-factor authentication. Due to the advancing distribution of mobile phones, speaker recognition plays a special role.

Despite all the advantages of biometrics, the vulnerability of the systems against attacks is still an existing weakness. In particular, speaker recognition systems are threatened. Due to the sophisticated research in the field of speech synthesis, a wide rage of effective methodologies for attacking speaker recognition systems is easy to utilize: i.e. replay, speech synthesis and unit-selection.

State-of-the-art countermeasures are successful in detecting synthesis and voice conversion attacks, but fail on detecting unit-selection attacks. The impact of these attacks to state-of-the-art speaker recognition systems is analysed. Thus, the focus of this thesis motivates: the creation and detection of unit-selection attacks, proposing a new countermeasure based on the principle of frequency analysis. For the evaluation of the experiments the metrics introduced in ISO/IEC CD2 30107-3 are utilized. Detection techniques of current research are discussed and new detection algorithms proposed. In contrast to conventional attack detection algorithms, which utilize feature extraction methods, known in the field of speech recognition for modelling the perception of sound by the human ear, the proposed algorithms disclaim the use of these filters and analyses the unfiltered frequency band.

By calculating a presentation attack score with the sum of the derivative in the frequency spectrum, an EER of 29.7% is yielded. An approach utilizing frequency-based features and machine learning techniques, i.e. SVMs and GMMs, improves the detection performance to 7.1% EER and 11.7% respectively on the unit-selection attacks of the Interspeech special session ASVspoof 2015.

ZUSAMMENFASSUNG

Biometrie gewinnt in der modernen Gesellschaft, insbesondere durch die zunehmende Verwendung der Mehr-Faktor-Authentifizierung, eine immer größere Bedeutung. Dabei wird die Sprechererkennung, gefördert durch die Verbreitung des Mobiltelefons, weiterhin eine Sonderrolle einnehmen.

Trotz aller Vorteile biometrischer Systeme ist die Verwundbarkeit der Systeme durch Angriffe ein bestehender Schwachpunkt. Die Sprechererkennung ist davon besonders betroffen, da hier aufgrund der fortgeschrittenen Forschung im Bereich der Sprachsynthese bereits effektive Angriffsmöglichkeiten existieren.

Aktuelle Methoden zur Angriffserkennung sind effektiv bei der Erkennung von synthetischen und stimmumformenden Angriffen, versagen jedoch bei der Erkennung von Unit-Selection Angriffen. Untersucht wird die Auswirkung dieser Angriffe auf Erkennungssysteme, welche dem aktuellen Stand der Technik entsprechen. Daher befasst sich diese Arbeit mit der Erstellung und Erkennung von Unit-Selection Angriffen. Die Auswertung der Experimente wird entsprechend der in ISO/IEC CD2 30107-3 vorgestellten Metriken durchgeführt. Erkennungsalgorithmen aus der aktuellen Forschung werden diskutiert und neue Methoden vorgeschlagen. Im Gegensatz zu herkömmlichen Algorithmen, welche zur Angriffserkennung Merkmalsextraktionsverfahren der Sprach- und Sprechererkennung verwenden, um die menschliche Wahrnehmung nachzubilden, analysiert der vorgestellte Algorithmus keine Filter, sondern berücksichtigt das gesamte Frequenzspektrum gleichermaßen.

Die Bestimmung eines Angriffswertes mittels Summe über die Ableitung des Frequenzspektrums zeigt eine Gleichfehlerrate von 29,7%. Werden Merkmalsvektoren der Frequenzanalyse mit Techniken des maschinellen Lernens kombiniert, so wird auf den Evaluationsdaten eine Gleichfehlerrate von 7,1% und auf den Unit-Selection Angriffen der Interspeech Special Session ASVspoof 2015 von 11,7% erreicht.

ACKNOWLEDGMENTS

First, I would like to thank my Supervisors *Prof. Dr. Christoph Busch* and *Andreas Nautsch* for the inspiring talks, motivation and knowledge. Thanks for leading me to the interesting field of biometrics and speech recognition. Further thanks go to *Prof. Dr. Nicholas Evans* and *Dr. Zhizheng Wu* for the obliging provisioning of data. Thanks to *Jessica Steinberger* for hours of support with Latex and R.

Special thanks go to my family, especially my parents, *Heike* and *Anton* for all the support and motivation during the writing of the thesis. Further thanks go to my college friend *Malte Hinrichs* for proofreading this document.

Finally I would like to thank my partner *Anna Müller* for all the love, patients and balance during the last years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

APCER	Attack Presentation Classification Error Rate
APIR	Attack Presentation Identification Rate
APMR	Attack Presentation Match Rate
APNRR	Attack Presentation Non-Response Rate
AT	Audiotory Transform
BPCER	Bona fide Presentation Classification Error Rate
BPNRR	Bona fide Presentation Non-Response Rate
CART	Classification And Regression Tree
CFCC	Cochlear Filter Cepstral Coefficient
CFCCIF	Cochlear Filter Cepstral Coefficient Instantaneous Frequency
dB	deciBel
DCF	Decision Cost Function
DCT	Discrete Cosine Transformation
DET	Detection Error Tradeoff
DFT	Discrete Fourier Transformation
DNN	Deep Neuronal Network
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transformation
EER	Equal Error Rate
EM	Expectation Maximization
FAR	False Acceptance Rate
FFT	Fast Fourier Transformation
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Rejection Rate
FSD	Full Scale Digital
FTA	Failure-To-Acquire
FTC	Failure-To-Capture

FTE	Failure-To-Enrol
FTX	Failure-To-eXtract
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
i-vector	intermediate-sized vector
IF	Instantaneous Frequency
ISO	International Organization for Standardization
JFA	Joint Factor Analysis
LDA	Linear Discriminant Analysis
LLR	Log Likelihood Ratio
LR	Likelihood Ratio
MAP	Maximum A-Posteriori
MFCC	Mel-Frequency Cepstral Coefficient
minDCF	minimum Decision Cost Function
ML	Maximum Likelihood
MLP	Mult Layer Perceptron
NFM	Near Field Monitor
PA	Presentation Attack
PAD	Presentation Attack Detection
PAI	Presentation Attack Instrument
PDF	Probability Density Function
PLDA	Probabilistic Linear Discriminant Analysis
PS-PD	PAD Subsystem Processing Duration
RMS	Root Mean Square
SIV	Speaker Identification Verification
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transformation
SVM	Support Vector Machine

TTS	Text-To-Speech
UBM	Universal Background Model
VAD	Voice Activity Detection
WCCN	Within Class Covariance Normalization

Part I

BIOMETRICS AND SPEAKER RECOGNITION

1 INTRODUCTION

1.1 MOTIVATION

Biometrics describes the biological and behavioural characteristic of an individual from which distinguishing, repeatable biometric features can be extracted for the purpose of biometric recognition [1]. The advantage of biometric recognition over common authentication methods, based on passwords of physical tokens, is the impossibility of losing, forgetting, or passing the biometric characteristic [2]. If someone acquires possession of the biometric feature of someone else, the characteristic and its features cannot be changed or replaced.

Important properties for the comprehensive use of a biometric system are the acceptability of the biometric capture process and the universality of the biometric characteristic that is used [2, 3, p. 15]. If, for example, the fingerprint utilised for identification, subjects with dermatological diseases may not be enrolled in systems which capture the structure of the upper dermal layer. If the biometric capture process is complicated, awkward or painful (inconvenient), it is less likely for the user to accept the biometric system.

The voice, as a biometric characteristic, is of particular interest as it is captured contactless and by low-cost sensors. In order to be robust against manipulation, further research is necessary. Especially the detection of replay and unit-selection attacks, which are based on the concatenation of speech units, is still crucial to state-of-the-art speaker recognition systems. Current research provides promising approaches for detecting synthesis and voice conversion attacks which can be followed up [4, 5, 6].

1.1.1 *Speaker Recognition*

Speaker recognition refers to determining subjects by their voice [7]. It can be used for identification and verification in security applications or for forensic scenarios. Unlike most biometric methods, which are based on analysis of images, speaker recognition utilises voice or speech recordings and features derived thereof. Such features can be extracted by several methods.

The voice as a biometric characteristic is gaining importance. Especially the flexibility and universality of the capturing process makes

speaker recognition interesting. Voice is a naturally produced signal [8]. It can easily be recorded with low-cost standard microphones under most conditions. In the case of mobile phones, most people, in particular the target group of business people, carry along a smartphone embedded microphone, which is designed for speech recording.

1.1.2 *Presentation Attack Detection*

Any biometric system is potentially vulnerable to attacks [9], thus a detection of attacks is required. As a common method for detecting replay attacks, liveness detection is used. Exemplary, for face recognition, analysis of the blinking can be applied to detect replays [10].

As any other biometric system, Speaker Identification Verification (SIV) systems are also vulnerable to attacks. As the field of speech synthesis is well studied, a broad range of effective attacks is available [11]. The synthesized voices of the subjects to incorporate are on such a high degree of quality and detail that conventional SIV algorithms are prone to falsely recognize attacks samples as genuine human, referred to as bona fide, samples. Especially with the objective of security-critical applications, further research is necessary for the purpose of ensuring a certain level of security. The vulnerability of speaker recognition systems for different attacks, such as replay [12] or speech synthesis [13] has been shown multiple times. There are approaches for detection and countermeasures, but they assume previous knowledge of the attack [14].

1.2 RESEARCH QUESTIONS

As the required performance of a biometric system has to be assessed by the operator, the research questions do not aim for a binary decision, but for an error rate for biometric systems in terms of Presentation Attack Detection (PAD) scores and moreover, the aim for implementation of reliable PAD subsystems. For the purpose of generating reproducible and harmonized error reporting, the metrics defined in ISO/IEC CD2 30107-3 [15] will be used. Following research questions will be more closely considered in thesis:

1. *To which extent are state-of-the-art SIV systems capable of rejecting unit-selection attacks in the absence of a PAD-system?*
   The performance of state-of-the-art SIV systems can be measured evaluating the Equal Error Rate (EER) of the Attack Presentation Match Rate (APMR) and False Non-Match Rate (FNMR).

2. *To which extend is an analysis of the frequency spectrum of speech samples capable of detecting unit-selection attacks?*

The accuracy of detection algorithms can be evaluated utilising Bona fide Presentation Classification Error Rate (BPCER) and Attack Presentation Classification Error Rate (APCER) for determining the EER and calculating a Detection Error Tradeoff (DET) function.

3. *What are the differences in using wavelet-transformation or spectrogram for unit-selection PAD?*
   In order to answer this question, the function of the wavelet-transformation and Short Time Fourier Transformation (STFT) will be examined.

1.3 ORGANISATION OF WORK

This thesis can be divided into three parts. The first part describes biometric systems in general and SIV systems more detailed. The second part provides a deeper understanding of attacks and PAD for SIV systems. In the last part, new detection algorithms are introduced, evaluated, and improved. The content is organized as follows:

Chapter 2: This chapter provides a technical overview. At the beginning, a global understanding of biometric systems is given. In the following the focus is set to a more detailed description of speaker recognition, including the mandatory signal processing theory and performance metrics. An introduction to speech synthesis and machine learning is given.

Chapter 3: An overview over attacks on biometric systems is given. Attacks on SIV systems are described in particular. Performance metrics for PAD are introduced.

Chapter 4: The developed PAD algorithms for unit-selection attacks are presented. After a description of the experimental set-up, the basic algorithms are evaluated. In a second step the basic algorithms are improved and evaluated again.

2

FUNDAMENTALS

2.1 BIOMETRIC SYSTEMS

In a general context, biometric systems can be understood as an identity management system. Every identity management system needs a method of establishing a person's identity. For this task, traditional systems employ knowledge-based (e.g. passwords) or token-based (e.g. keys or ID cards) methods. In contrast, biometric systems acquire biometric data from an subject and extract features from the acquired data in order to compare it with a database of enrolled biometric samples [2].

2.1.1 *Topology*

Independent of the modality used by the biometric system, the system can be divided into five subsystems. Each subsystem is mandatory and has a specific role, which is specified in [16] and depicted in figure 1.

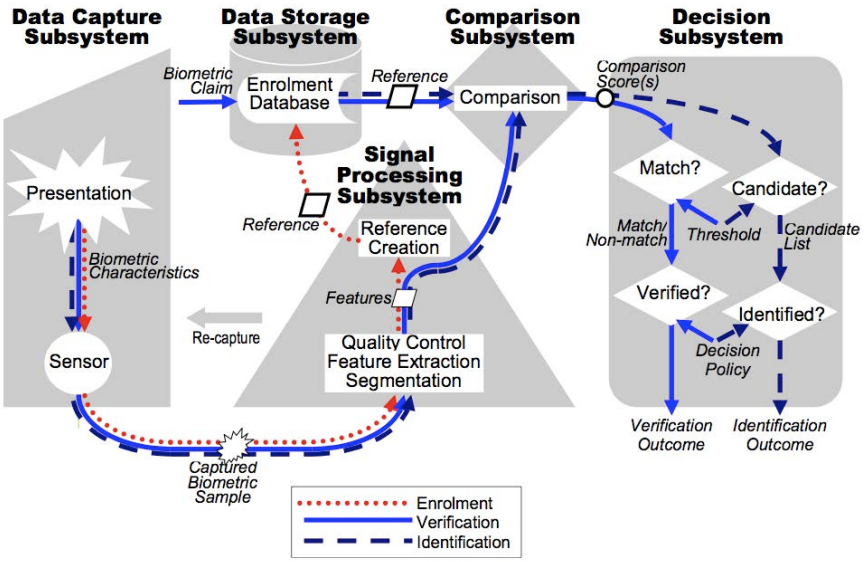


Figure 1: Topology of a biometric system, according to [16]

Data Capture Subsystem: This step converts the biometric characteristic into a captured biometric sample. A sensor e.g., microphone, converts the represented characteristic e.g., voice, into an electric signal. Also the concomitance of multiple sensors is possible. The captured biometric sample is passed to the signal processing subsystem.

Signal Processing Subsystem: The further processing of the captured biometric sample takes place in the signal processing subsystem. The features are extracted from the biometric sample. Depending on the algorithm, a preceding segmentation is needed; some systems incorporate a quality control mechanism. Either the created features are enrolled in the database as reference or they are used as probe for the comparison.

Data Storage Subsystem: The references of enrolled subjects are managed in the enrolment database of the data storage subsystem. Generally, the tasks of the data storage subsystem are inserting new references into the enrolment database (enrolment), returning all references (identification), or returning a specific reference belonging to a biometric claim (verification).

Comparison Subsystem: The reference, given by the data storage subsystem, and the probe, generated by the signal processing subsystem, are compared in the comparison subsystem. This system produces a comparison score for each compared reference. The comparison score can be available as similarity score or dissimilarity score.

Decision Subsystem: The last stage in identification and verification is the decision subsystem. In order to receive a binary decision, the scores of the comparison subsystem are assessed by means of a threshold or other decision policies. The result is a binary decision if a subject is possibly verified or identified.

2.1.2 *Operation Modes*

According to ISO/IEC 19795-1 [16], there are three distinct operation modes for biometric systems: enrolment, verification, and identification.

Biometric enrolment: This process refers to the registration a new reference to the enrolment database. This step is mandatory, as both, verification and identification, are comparing a probe against references stored in a enrolment database. The biometric characteristics are captured as a biometric sample via the data capture subsystem. In order to obtain a reference, the features are extracted in the signal processing subsystem. For enrolling a new reference, comparison and

decision subsystems are not required. Subjects are enrolled utilizing one or multiple reference samples for the purpose of creating reference templates or reference models, respectively. Where templates represent a *reference biometric feature set*, and models represent function generated from biometric data [16].

Biometric verification: The process of confirming a biometric claim through biometric comparison is called biometric verification [17]. By means of the data capture subsystem and the signal processing subsystem, features are derived from the biometric characteristics and passed to the comparison subsystem as a probe. According to the biometric claim, the database storage subsystem passes the reference, matching to the claim, to the comparison subsystem. The comparison subsystem processes probe and reference, receiving a comparison score. The score is assessed according the defined threshold and the binary decision, whether the subject is verified or not, can be made. Typical usage for biometric verification are access control systems, like border control systems or other mechanisms to ensure no unauthorized usage of protected resources.

Biometric identification: Biometric identification refers to the process of determining the biometric reference identifier associated to an individual. In this case, a biometric reference identifier describes a pointer to a certain reference in the enrolment database [17]. As in the verification process, the probe is provided by the data capture system and signal processing subsystem. Unlike the verification process, the identification process calculates the comparison scores for all references available in the enrolment database to the probe. Based on the comparison scores, the decision subsystem can make the decision, whether the individual is identified or not. A typical use case for biometric identification is represented by forensics, for example identifying flood victims, but also the usage for blacklists, like casino-blacklists for gambling addicted, is possible as well.

Further system advantages: Furthermore, biometric systems offer advantages over common identity management systems. Due to the difficulty in replacing, exchanging, or losing biometric characteristics, a biometric system is able to determine negative recognition and non-repudiation [18]. The capability of asserting if a subject is already enrolled in the system is referred to as de-duplication. Biometrics can be used as a proof of identity for receiving social welfare or retirement pension. In 2014, India started a large biometric database called AADHAAR[1]. This database gives every citizen of India a proof of identity which enables them to receive state aid. De-duplication ensures that no citizen obtains a second identity.

1 http://uidai.gov.in

As it is almost impossible to replace or alter biometric characteristics, the biometric system assures that a identified or verified person can not disclaim the recognition afterwards [19, p. 2]. This feature, referred to as non-repudiation, is important for forensic scenarios, for example a criminal convicted by means of his fingerprint.

2.1.3 *Performance Estimation*

In general, the outcome of a conventional identity management system depending on passwords or tokens is linear dependent on the input. If a password is entered or a token presented in order to access a resource, it is either accepted or not. For biometric systems however, the outcome depends on lots of factors, which can lead to errors. The following different types of errors which may arise during the biometric processing are standardized in [16] and [17].

Failure-To-Capture (FTC): The proportion of failures of the biometric capture process to produce a captured biometric sample [17]. This describes failures that occur in the data capture subsystem. The FTC can be calculated as:

$$\mathrm{FTC} = \frac{N_{tca} + N_{nsq}}{N_{tot}}, \tag{1}$$

where N_{tca} is the number of terminated capture attempts, N_{nsq} the number of images with insufficient sample quality and N_{tot} the total number of capture attempts.

Failure-To-eXtract (FTX): The proportion of failures of the feature extraction process to generate a template from the captured biometric sample, N_{ngt}, to the number of successful captured samples, N_{sub}. This describes failures that occur in the signal processing subsystem. The FTX can be calculated as:

$$\mathrm{FTX} = \frac{N_{ngt}}{N_{sub}}. \tag{2}$$

Failure-To-Enrol (FTE): The proportion of a specified set of biometric enrolment transactions that resulted in a failure to create and store a biometric enrolment data record, N_{nec}, to the total number of subjects, intended to be enrolled in the biometric application, N [17]. This describes failures that occur in the data storage subsystem. The FTE can be calculated as:

$$\mathrm{FTE} = \frac{N_{nec}}{N}. \tag{3}$$

Failure-To-Acquire (FTA): The proportion of a specified set of biometric acquisition processes that were failure to accept for subsequent comparison of the output of a data capture process [17]. This metric summarizes the failures of the data capture subsystem and the signal processing subsystem. The FTA can be calculated as:

$$FTA = FTC + FTX \cdot (1 - FTC). \tag{4}$$

FNMR: Proportion of genuine attempt samples falsely declared not to match the template of the same characteristic from the same subject supplying the sample [16]. This failure occurs in the algorithm of the comparison subsystem. The FNMR for a specific threshold t can be calculated as:

$$FNMR(t) = \int_0^t \Phi_g(s)ds, \tag{5}$$

where $\Phi_g(s)$ represents the Probability Density Function (PDF) of the genuine comparisons with s as similarity score.

False Match Rate (FMR): Proportion of zero-effort impostor attempt samples falsely declared to match the compared non-self template [16]. This failures occur in the algorithm of the comparison subsystem. The FMR for a specific threshold t can be calculated as:

$$FMR(t) = \int_t^1 \Phi_i(s)ds, \tag{6}$$

where $\Phi_i(s)$ represents the PDF for the imposter comparisons, with s as similarity score.

The two metrics, FNMR and FMR, describe comparison algorithm errors. In order to determine the overall performance of a biometric system further metrics are needed:

False Rejection Rate (FRR): The proportion of verification transactions with truthful claims of identity that are incorrectly denied [16]. The FRR can be calculated as [1]:

$$FRR = FTA + FNMR \cdot (1 - FTA). \tag{7}$$

False Acceptance Rate (FAR): The proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed [16]. The FAR can be calculated as [1]:

$$FAR = FMR \cdot (1 - FTA). \tag{8}$$

The following metrics are not defined in International Organization for Standardization (ISO) documents, but common in the field of the evaluation of biometric systems [3, 20].

EER: The point where error rates are equal. Commonly FMR and FNMR are compared. In the field of PAD, BPCER and APCER are utilised instead.

FMR100: The FNMR when FMR is 1%.

FMR1000: The FNMR when FMR is 0,1%.

2.2 SPEAKER RECOGNITION

Most SIV systems rely on a two-staged system. First, the features are extracted from the digital speech signal, conventionally referred to as front-end. In the second stage, the extracted features are processed to rather biometric features and used for decision making. This stage is conventionally referred to as back-end.

2.2.1 *Front-End*

In SIV systems, there are several methods for extracting the features. They can be divided into at least three categories [7]:

High-level features: Features extracted from the used vocabulary and phrases of the speaker are referred to as *high-level features*. This method of extracting features requires a complex preprocessing, as the spoken text has to be understood and interpreted. For the training process a large set of training data is needed [21, 22].

Prosodic features: The second approach is the recognition and comparison of voice timbre and rhythm of the speaker. According to [23], Hidden Markov Models (HMMs) or Support Vector Machines (SVMs) are used for classifications.

Low-level features: The most common way of feature extraction is the analysis of the frequency spectrum. In order to transform the speech signal into the frequency domain by utilization of discrete Fourier transformation, the speech signal usually is segmented into parts with a duration of 20 ms to 30 ms with a step size of 15 ms. Applying a window function to these parts reduces the error caused by the Fourier transformation. In the frequency domain, the signal can be influenced by filter banks, e.g. the Mel-scale [7]. Mel-scale based features representing spectral properties are called Mel-Frequency Cepstral Coefficients (MFCCs).

2.2.2 *Back-End*

Further processing and decision making is commonly done by methods of machine learning. In speaker recognition, SVMs or Gaussian Mixture Models (GMMs) are well established [24]. Different microphones or changing background noises cause channel effects which lower the performance of the system. *Factor analysis* is a commonly used method, i.e. in the Joint Factor Analysis (JFA) approach, the GMM-supervector is decomposed into speaker factors, channel factors, and residuals [25].

State-of-the-art algorithms extract Baum-Welch sufficiency statistics from the Universal Background Model (UBM), which is a GMM. The statistics are represented in intermediate-sized vectors (i-vectors) [26], a JFA special case which describes a voice sample's offset factor from the UBM. i-vector features are projected into a biometric-discriminant, unit-sphere space by Linear Discriminant Analysis (LDA), Within Class Covariance Normalization (WCCN) and radial gaussianization, i.e. length normalization [27]. State-of-the-art comparisons are conducted by Probabilistic Linear Discriminant Analysis (PLDA).

2.2.3 *Metrics for Speaker Recognition*

Likelihood Ratio (LR): Likelihood is defined as $\Pr(x|y)$, where x represents the hypotheses and y the observation. In order to assess a sample, the likelihood of the sample (observation) belonging to genuine or impostor (hypotheses) is determined. The ratio of both hypotheses is given as the probabilities of genuine hypothesis H_0 given the observation E over the probability of impostor hypothesis H_A given the same observation. The required posteriori probability of a hypotheses given E is unknown. Using the Bayes-Theorem, the likelihood ratio can be calculated [28]:

$$\frac{P(H_0|E)}{P(H_A|E)} = \frac{P(E|H_0)}{P(E|H_A)} \times \frac{P(H_0)}{P(H_A)}. \tag{9}$$

The left part of the equation also is referred to as *a-posteriori odds*. The ratio of the probabilities of the hypotheses depicts, whether a hostile of friendly scenario is more likely. A common definition of the prior is the population prior, which can be constrained to how many attacks will the system face on average. The ratio is referred to as *a-priori odds*, represented by π and has to be defined for the system. The ratio of the probability of the evidence in genuine or impostor is called *likelihood ratio*. A likelihood ratio score S is calculated as:

$$S = \frac{P(E|H_0)}{P(E|H_A)}, \tag{10}$$

such that the posteriori probability of the genuine hypothesis can be reformatted as:

$$P(H_0|E) = \frac{S\pi}{1 + S\pi}. \tag{11}$$

Log Likelihood Ratio (LLR): Logarithmic presentation of the LR. Simplifies the calculation and definition of a Bayesian threshold.

Cost of LLR (C_{llr}): Measure for the goodness of LLR scores. The C_{llr} is calculated by integrating over the prior-weighted FNMR and FMR, where the priors represent operating points/thresholds η.

minimum Cost of LLR (C_{llr}^{min}): Since systems may not be perfectly calibrated, the C_{llr}^{min} estimates the goodness of the perfectly calibrated system. The difference between C_{llr}^{min} and C_{llr} is called calibration loss; it is a measure for the possible improvement in making better decisions on average by calibration. C_{llr} defines the discrimination loss, which is caused by the algorithm.

minimum Decision Cost Function (minDCF): The minDCF is a metric representing the minimum of the Bayesian as a Decision Cost Function (DCF) error-rate for an application specific prior π.

2.3 SPEECH SYNTHESIS

In general, speech synthesis describes the backend of Text-To-Speech (TTS) systems, whereas the frontend is the part of text and linguistic processing [29]. The backend receives the phoneme-based information and transforms it context-sensitively into audible speech [20, p. 1382]. There are several methods for this transformation. This section describes the most common according to [29] and [20].

2.3.1 *Articulatory Synthesis*

Articulatory synthesis tries to emulate the production process of human voice with mathematical models. The basic oscillation of human

voice, the root chord, is an interaction of the mechanical vibrations of the vocal chord and the airflow of the lungs. The root chord passes the vocal tract with tongue, jaw, lips, and nose which can be represented as linear or non-linear acoustic filters, whereby the human voice develops. As the mathematical models have to be highly accurate, the speech synthesized by articulatory synthesis is of low subjective quality. Therefore this method is most likely suitable for research [20, p. 1384].

2.3.2 *Formant Synthesis*

During the production of human speech, certain frequencies of the root chord are filtered, other frequencies gain by resonances. Formants are the frequency-ranges with the highest gain. The formant with the lowest frequency is referred to as f0 or pitch. The formant synthesis utilizes the formants to represent the human vocal tract with simple filters. For each voice to be synthesized, a convenient waveform generator is needed which emulates the vocal chord. Subsequently the signal is filtered by simple filters.

Computational and storage requirements are low, this method is common for speech synthesis with embedded applications, like mobile phones or navigation systems. The system delivers synthetic speech with high quality replication of the consonants [20, p. 1384].

2.3.3 *Concatenative Synthesis*

In general, concatenative synthesis describes methods where recorded snippets of human speech are reassembled to new utterances.

There are many different possibilities for selecting the size of the speech units. A straightforward approach would be to use words. As the English language contains at least 170 000 words [30], the database for an universal word based concatenative synthesis would be enormous. For high-quality synthesis, a single representation of each word would not be enough, as the several versions of the word have to be recorded for different contexts [20].

In order to reduce the needed storage space, commonly smaller units are utilized, e.g. syllables or even phonemes. The English alphabet contains 26 letters. As some of the letters can be used with variations and combinations, British English comprises a total of 44 phonemes. The syllables are concatenations of phonemes, so the number of combinations is accordingly higher. Short units offer the advantage of a smaller need of storage space, but speech synthesized with short elements sounds artificial and choppy [31]. In order to obtain less

choppy transitions between units, state-of-the-art systems employ diphones. A diphone is a short speech segment, reaching from the middle of one phone to the middle of the next phone. The cut points are located in the acoustic most stable region. The English language consists of at least 1000 diphones, which can vary depending on the prosody [32].

The main issue for a human sounding and smooth concatenation synthesis is the appropriate selection of the units and smooth transitions between the units. In conventional approaches, possible concatenations of units are calculated in advance [29]. An advanced method is the online synthesis, where the optimal unit is selected during the synthesis process. Unlike the offline calculation, where only general static informations are available, the online calculation has access to the full context information [29]. Pre-processing the vast amount of contextual information is rather expensive, where latest synthesis approaches benefit from technological advances in terms of computational power, such that optimal fitting units can be found under effortable costs. Moreover, different unit types, like phonemes, diphones, and syllables can be utilised for the synthesis of one sentence. This is called unit-selection.

2.3.4 *Unit-Selection Synthesis*

A fast model for an online selection of units is essential for the manifold opportunities of selecting the appropriate unit. This section depicts the unit-selection, introduced in [33]. In order to enable fast searching on the data, a vector of features is assigned to each unit. The features used can vary between different systems. Typical utilised features are phoneme label, duration, signal power, and the first formant, but also acoustic features. In order to estimate convenient units, a distance measurement is required for each feature. For continuous features, as signal power or duration, common distance metrics can be utilised, the distance for discrete features needs to be defined. The feature vectors are computed offline.

In order to select the most convenient unit with minimal distortion, two distance metrics are used:

Target Cost: Also referred to as unit segmental distortion, this metric defines the difference $D_u(u_i, t_i)$ between the selected unit vector $u_i = \{uf_1, uf_2, ..., uf_n\}$ and the target unit $t_i = \{tf_1, tf_2, ..., tf_n\}$. The vectors need to be of the same type. In order to mitigate or boost specific features, the resulting cost vector can be weighted by a weight vector $W_u = \{w_1, w_2, ..., w_n\}$. In a realistic scenario, the target unit is unknown. In order to be able to calculate the target cost, a hypothet-

ical target has to be used which is derived from the sentence to be synthesized [29].

Join Cost: Also referred to as unit concatenative distortion, this metric defines the difference $D_c(u_i, u_{i-1})$ between unit u_i and its previous adjoining unit u_{i-1}. The resulting vector D_c represents the cost for joining these two units. The cost can be weighted by the weight vector for join cost W_c. In contrast to the target cost, the join cost can be directly computed with the candidate units. Figure 2 visualizes the origin of target cost and join cost.

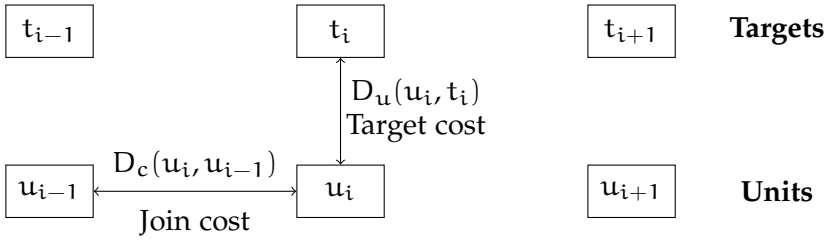


Figure 2: Costs for the selection of a unit [33]

The overall cost of a completed unit sequence is defined over the weighted sum of the costs of each unit [33]:

$$\sum_{i=1}^{n} \left(D_c(u_i, u_{i-1}) \cdot W_c + D_u(u_i, t_i) \cdot W_u \right). \tag{12}$$

n is the number of segments in the target utterance. W_c and W_u are weights for influencing the relevance of target cost and join cost. In order to find the best unit sequence, equation 12 has to be minimized.

2.4 SIGNAL PROCESSING

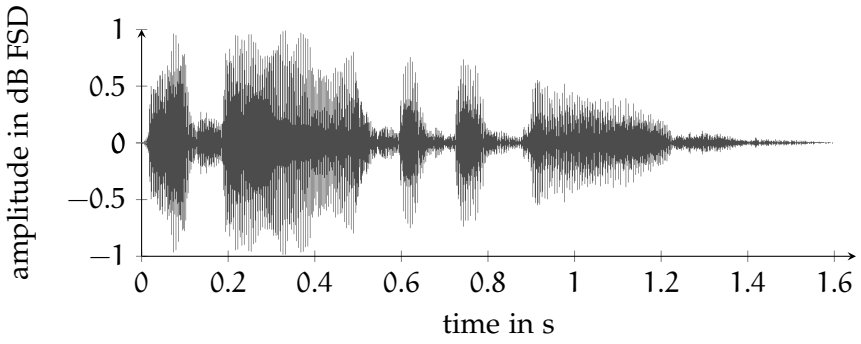
For most biometric modalities, a preprocessing of the biometric samples improves the performance of the biometric system. In order to generate comparable features, most image processing feature extractors require a proper segmentation and alignment of the image. For example, for iris recognition, the iris has to be detected and cropped in order to encode the structure of the iris [34].

Even though speaker recognition is not based on image processing, a preprocessing of the signal is useful. In general, audio data can be represented as a waveform signal. In order to digitise the signal, the information has to be discretised. The accuracy of the sampling depends on the used coding. For example, a 16 kHz codec is capable of storing twice the information of a 8 kHz codec.

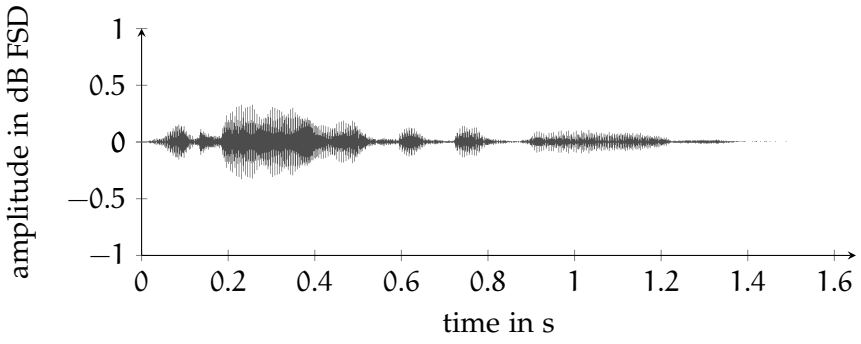
2.4.1 *Amplitude Normalization*

For signals represented as waveform, the amplitude describes the peaks of the oscillation. For digital audio signals the maximum amplitude is limited by the highest representable value of the encoding, also referred to as clipping point or 0 deciBel (dB) Full Scale Digital (FSD).

One audio signal recorded by different devices can be represented with different amplitudes. Figure 3 displays the waveform of a speech signal recorded by a Samsung Galaxy Note 4 (figure 3a) and an LG G2 (3b). The mobile phones were placed close to each other, so the energy of the signal, reaching the microphone, was approximately identical. The difference between the signals is caused by the microphone and the software processing the signal.



(a) Signal recorded with a Samsung Galaxy Note 4



(b) Signal recorded by an LG G2

Figure 3: Waveform representation of captured speech signal

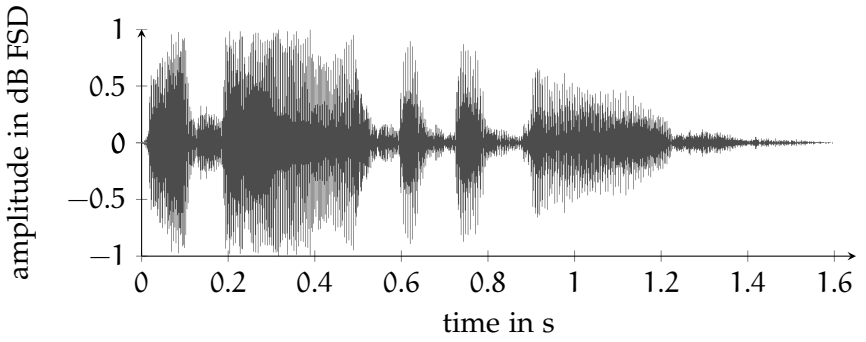
In order to achieve comparable signals, the recordings can be normalized. Normalization can be understood as a level adjustment of the signal. Two common ways for normalization are maximum normalization and Root Mean Square (RMS) normalization.

Maximum normalization: The factor for adjusting the highest value of the signal to the clipping point is determined. Then, the complete signal is multiplied with this factor. For some circumstances it is useful to avoid the clipping point. This can be done by multiplying the factor with a constant < 1 [35].

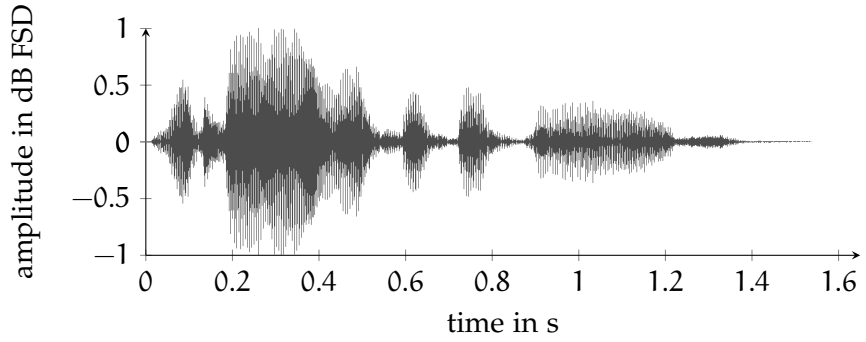
As the highest value of the signal is set to the clipping point or lower, the maximum normalization assures the avoidance of clipping. Due to the usage of a single point, a signal error or peak can strongly influence the normalization.

RMS normalization: The RMS normalization utilizes the RMS value of the signal instead of the maximum. This approach is more robust against outliers, but harbours the risk of exceeding the clipping point.

Figure 4 shows the speech signals of figure 3 after an RMS normalization. Differences in the signal are caused by the different microphone characteristics used in the smartphones.



(a) Signal recorded with a Samsung Galaxy Note 4, normalized



(b) Signal recorded with an LG G2, normalized

Figure 4: Waveforms of the normalised samples displayed in figure 3

2.4.2 *Voice Activity Detection*

The success rate of most signal processing algorithms is highly dependant on the quality of the signal processed. Voice Activity Detection (VAD) is the task of locating the speech signal inside an audio signal. For robust speaker recognition systems, VAD is an essential operation [36]. There are different approaches of detecting speech. A few algorithms are described below.

Zero Crossing Measure: A possibility is to count the number of changes in the signum of the signal. By counting, the crossing of zero values can be computed. In general, the zero crossing rate of speech is lower than the zero crossing rate of unvoiced frames [37]. In order to distinguish the frames, a threshold for the zero crossing value has to be chosen. Zero Crossing Measure is easy to implement and fast in computing, but only suitable if the background noise does not cause zero crossings.

Energy-based VAD: Energy-based VAD is the most common VAD in speaker recognition [36]. It is a simple solution that performs best in noise-free conditions. First, the signal is normalized and segmented into frames. The signal energy $P(i)$ of every frame i is calculated. Afterwards, the maximum energy of the frames is determined as [38]:

$$P_{max} = max\left(P(j)_{j=1,2,\dots,N}\right). \tag{13}$$

In order to distinguish between speech and non-speech frames, the energy of every frame is assessed as follows:

$$i = \begin{cases} \text{speech}, & \text{for } P(i) \geqslant T_{min}, \\ \text{non-speech}, & \text{for } P(i) < T_{min}. \end{cases} \tag{14}$$

T_{min} is a threshold which has to be selected in dependence from P_{max}. Common values for T_{min} are between 30 and 48 dB [7, 39]. This method works well for signals with a high Signal to Noise Ratio (SNR), but for noisy environments a lot of non-speech frames are selected as speech. To improve the results of the energy based VAD, speech enhancement methods can be applied, in order to increase the SNR [36].

2.4.3 *Frequency Analysis*

In the field of signal processing and analysis, the examination of the frequency spectrum of the signal is one of the most important processes. There are different approaches to obtain the frequencies incorporated in a signal.

Fourier transformation: Every integrable continuous-time signal can be interpreted as a superposition of harmonic oscillations. Periodic signals are composed of a basic oscillation, overlain by harmonic components. In general, the signal can be represented as a sum of oscillations [40, p. 923]:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cdot \cos(n\omega_0 t) + b_n \cdot \sin(n\omega_0 t)], \tag{15}$$

whereas ω_0 represents the frequency of the basic oscillation and $n\omega_0$ are the equivalent frequencies of the harmonic components. The coefficients a_n and b_n can be interpreted as the amplitudes of the associated harmonic frequencies.

The Euler equation enables a representation of trigonometric functions as a linear combination of imaginary exponential functions by utilizing the unit-circle. $\sin(x)$ and $\cos(x)$ can be converted as:

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2j}, \quad \cos(x) = \frac{e^{jx} + e^{-jx}}{2}. \tag{16}$$

Using the Euler equation, the decomposed signal can be represented in the shorter complex form:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{jn\omega_0 t}. \tag{17}$$

As in equation 15, ω_0 describes the frequency of the basic oscillation, $n\omega_0$ the frequencies of the harmonic components. Contrary to the trigonometric representation the complex representation is defined by c_n, which is the amplitude of the oscillations. As discrete frequencies are used for decomposing the signal, the resulting frequency spectrum is also discrete.

The presented representation can only be used for periodic oscillations [41]. In general, a periodic function has a periodic time T in which the oscillation repeats. For example, $\sin(x)$ repeats every 2π. The periodic time T is defined with $T = \frac{2\pi}{\omega_0}$; for the sinus function this leads to $T = 1$. In a realistic scenario most signals are non-periodic. For periodic oscillations with a larger periodic time, T has to be increased accordingly. In order to analyse non-periodic signals, the periodic time of the frequency can be interpreted as infinite [41]. The

gap between two frequencies in the frequency spectrum $\Delta\omega$ can be calculated as:

$$\Delta\omega = \omega_{n+1} - \omega_n = (n+1)\omega_0 - n\omega_0 = \omega_0 = \frac{2\pi}{T}, \tag{18}$$

with $T \to \infty$ equation 18 converges to zero. The frequency spectrum is no longer discrete, but continuous. Then, the sum in equation 17 transforms into an integral:

$$f(t) = \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} F(\omega) \cdot e^{j\omega t} d\omega. \tag{19}$$

Equation 15 and 17 accumulate the basic frequency ω_0 and the multiples of it, whereas the associated frequency is defined by a coefficient. For the continuous representation, the amplitude for each frequency can be calculated by $\frac{1}{2\pi}F(\omega)$. $F(\omega)$ is referred to as Fourier transform of $f(t)$ [41]. In order to obtain the Fourier transform, equation 19 can be rearranged to:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt. \tag{20}$$

Discrete Fourier Transformation (DFT): As the calculation of the infinite integral is time consuming or impossible, the conventional Fourier transformation is non-satisfying for application in computational analysis of signals. Additionally, a signal (e.g. video or audio) captured by a computer has to be sampled in order to obtain computable data. This leads to discrete data, whereby the infinite integral becomes impractical. For computation of discrete non-periodic signals, the **DFT** can be used. It is assumed that a restricted period from 0 to $N-1$ of the signal is of interest [41]. Furthermore, a discrete sampling of the signal is presumed. The sampling interval is defined by T_a, the moment of the samples are $n \cdot T_a$, with n between 0 and $N-1$. The overall sample time of the signal can be calculated as $N \cdot T_a$. With these requirements, equation 17 can be adapted with $t \to n \cdot T_a$, resulting in [40, p. 925]:

$$F_d(k) = \sum_{n=0}^{N-1} f(nT_a) \cdot e^{\frac{-j2\pi kn}{N}}. \tag{21}$$

The continuous angular frequency ω is substituted by the discrete term $\frac{2\pi k}{N}$, with $k = 0, 1, ..., N-1$.

Fast Fourier Transformation (FFT): In order to decrease the computation time, the **FFT** can be used. The basic idea of the FFT is to split the signal $f(t)$ into N equal sized elements, calculating the DFT of each element and merging the results by superposition. The best performance can be achieved by splitting the signal into 2^k elements. First,

the signal is divided into two elements. These two are divided again, and so on. After the calculation of the DFT, the elements are merged level by level [40, p. 925].

STFT: The Fourier transformation transforms the representation of the signal from time domain to frequency domain. Due to this transformation, the assignment of frequencies to a certain time slot is not possible. For the analysis of audio signals, a transformation with respect to the time-domain is needed. The STFT is a common tool for the time-frequency analysis [42, p. 285ff].

The Fourier transformation serves as a basis. The signal $f(t)$ is multiplied by a window function defined by $\gamma(t-\tau)$. τ enables a sliding of the window in time. γ can be represent different window functions. In order to generalise, γ^* will be used as substitution for all window functions. The time-dependent STFT, introduced in [43], can be derived from equation 20:

$$F(\tau,\omega) = \int_{-\infty}^{\infty} f(t)\gamma^*(t-\tau)e^{-j\omega t}dt. \tag{22}$$

The new variable τ adds the time-dependency to the transformation. As $\gamma^*(t-\tau)$ represents a window function, the signal outside of the window is suppressed.

The introduced STFT is designed for a continuous signal. For applications in digital signal analysis, an adoption of the equation is necessary, as the signals are sampled and therefore discrete. Similar to the DFT, the STFT can be rearranged, substituting the integral with a sum:

$$F^{\gamma}(m,k) = \sum_{n} f(n)\gamma^*(n-mN)e^{\frac{-j2\pi kn}{N}}. \tag{23}$$

γ^* can be interpreted as transformation core, which is modulated by $e^{\frac{-j2\pi kn}{N}}$. For a fast calculation of the sum, FFT can be employed [42, p. 293f].

Spectrogram: The consideration of time during the transformation adds a new dimension to the result. Whereas the Fourier transformation results in a representation of frequency and amplitude, STFT transforms the signal to a representation in frequency, amplitude and time. In order to illustrate the transformation, the so called spectrogram is used. In most cases the results of STFT are complex. In order to simplify the representation, the square of the absolute value is calculated as [42, p. 290]:

$$S_x(\tau,\omega) = |F_x^{\gamma}(\tau,\omega)|^2. \tag{24}$$

$S_x(\tau, \omega)$ is plotted as spectrogram, where time (τ) is displayed at the x-axis, frequency (ω) at the y-axis, and the amplitude (S_x) is pictured as grey scale.

Wavelet Transformation: An alternative to STFT is the wavelet transformation. It is known to be faster than the FFT [44]. The main difference between STFT and the wavelet transformation is that the transformation cores of the wavelet transformation (the mother wavelet function) is scaled and not modulated. The wavelet in the wavelet transformation is defined as $\psi^*\left(\frac{t-b}{a}\right)$, where b translates and a dilates the wavelet. The wavelet transformation is comparable to the continuous STFT, equation 22, but the modulated transformation core is replaced by the scaled transformation core:

$$W(b, a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) dt. \tag{25}$$

The prefix $|a|^{-\frac{1}{2}}$ effects that all equations $|a|^{-\frac{1}{2}}\psi(\frac{t}{a})$ obtain the same energy. It can be interpreted as prefix of the transformation core.

In general, the wavelet function can assume every form, as long as it enables an error-free back transformation [42, p. 302]. This requirement is assured as long as

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty. \tag{26}$$

is fulfilled. $\Psi(\omega)$ is the Fourier transformed of $\psi(t)$. In order to let equation 26 become true,

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0 \tag{27}$$

has to be true as well and $|\Psi(\omega)|$ for $|\omega| \to 0$ and $|\omega| \to \infty$ has to be fading [42, p. 302].

Discrete Wavelet Transformation (DWT): For discrete values, the wavelet transformation can be accelerated. Due to equation 26, the wavelet can be interpreted as a bandpass filter. If the wavelet is dilated, the bandpass is translated in the frequency domain.

According to the Mallat theorem [45], a successive decomposition of a signal into bandpass-signals is possible without losing information. Figure 5 visualizes the idea of the DWT.

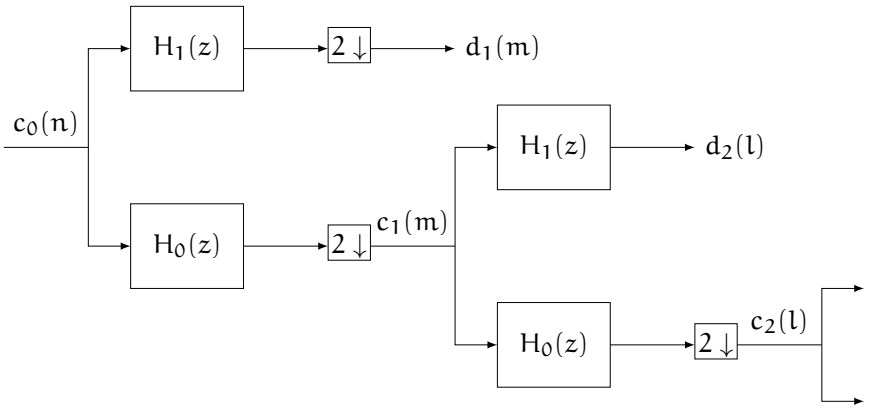


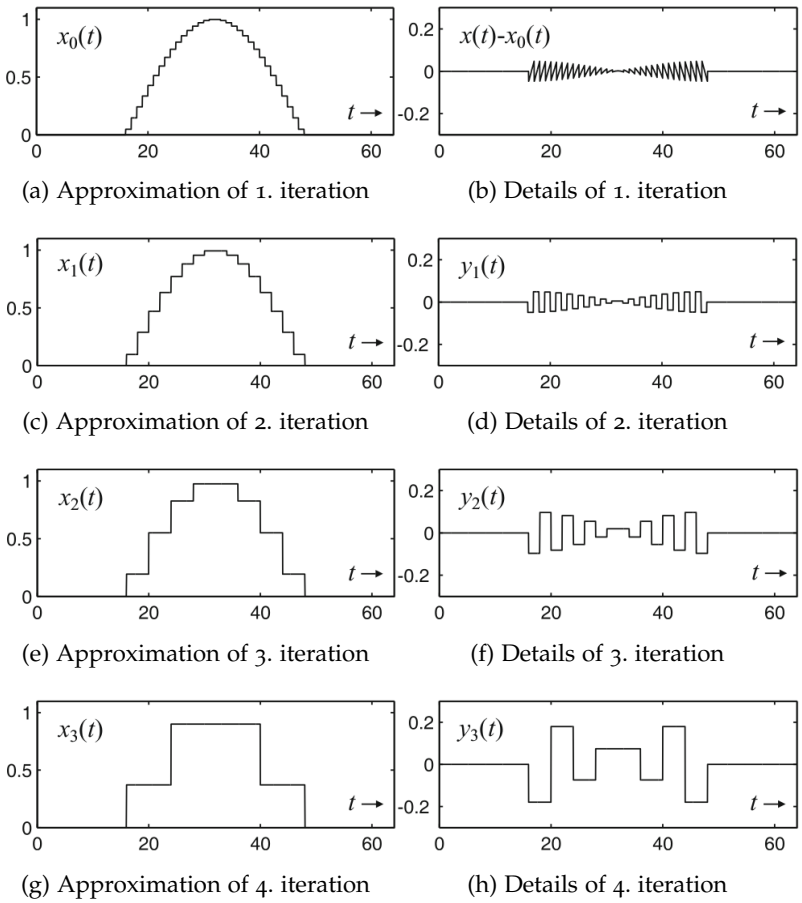
Figure 5: Filterbank for DWT calculation [42]

u $c_0(n)$ represents the signal. $H_1(z)$ displays a high pass filter, $H_0(z)$ has to be the inverse of $H_1(z)$, a low pass filter. Only every second sample of the outgo of the filters is kept, indicated by $2\downarrow$. This result of the high pass filter is the first detail level $d_1(m)$, the result of the low pass filter is the first approximation $c_1(m)$. In the next step, the approximation $c_1(m)$ can be decomposed by the same filter, as the sample rate is reduced in the previous step.

An example of a DWT is given in figure 6. A sine half wave is decomposed using a Haar wavelet [46]. The right side displays the details, the left side the approximation.

Difference between STFT and DWT: The difference between both tools is the resolution of time and frequency. The STFT has a uniform time resolution over all frequencies. The signal is divided into uniform slots, in which each frequency is determined. A larger time slot provides a more precise determination of the frequency domain, but the time-domain resolution is poorer. Smaller timeslots however provide a better resolution in time, but a poorer resolution of the frequencies. Figure 7a shows the uniform resolution of time and frequency for all frequencies.

As the wavelet in the wavelet transformation is translated, the size of the analysis window changes between the frequencies. The wavelet used for analysing high frequencies is smaller than the transformed window used for low frequencies. The time accuracy for higher frequencies is more precise and the frequency accuracy is less precise [47].



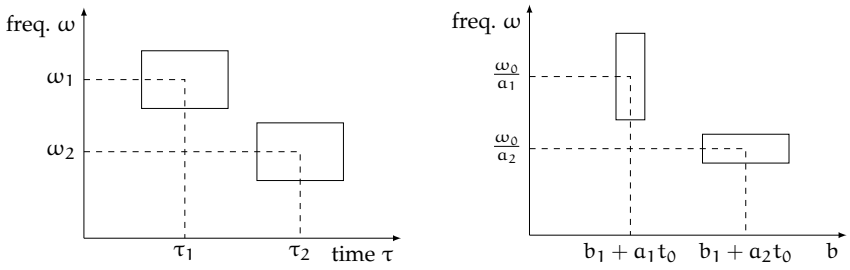
(a) Approximation of 1. iteration (b) Details of 1. iteration

(c) Approximation of 2. iteration (d) Details of 2. iteration

(e) Approximation of 3. iteration (f) Details of 3. iteration

(g) Approximation of 4. iteration (h) Details of 4. iteration

Figure 6: DWT transformation of a sine half wave with a Haar-wavelet [42]



(a) STFT (b) Wavelet Transformation

Figure 7: The relation of time and frequency resolution for STFT 7a and wavelet transformation 7b [42]

2.5 MACHINE LEARNING

In a global context, machine learning describes a type of algorithm that learns from past experience to make decisions [20]. It can be classified in the field of computational intelligence. There are different fields of machine learning, all utilize the theory of statistics for generating mathematical models [48]. Machine learning can be divided into at least two main classes: descriptive and predictive algorithms. Descriptive algorithms, e.g. clustering algorithms, aim to extract new information from data. Predictive algorithms aim to learn dependencies between populations, in order to predict the population of new data. Since in the field of biometrics mainly predictive algorithms are used, descriptive algorithms are not considered any further.

There are lots of machine learning theories and algorithms. Due to timing constraints this thesis will focus on only two: SVM and GMM. SVMs are well-examined for binary classification and pattern recognition [20, p. 1510]. GMMs are often used in speaker recognition, as they are capable of modelling a large variability of sample distributions.

2.5.1 *Support Vector Machines*

The SVM is a binary, linear classifier that separates the space into two regions by a hyperplane. The basic idea is, that the SVM selects a hyperplane, that provides the best generalization capacity [20, p. 1505]. Figure 8 illustrates a simple example of a two dimensional SVM. Population 1 and Population 2 are divided by the linear hyperplane. The nearest data points to the hyperplane are referred to as support vectors, the distance from the support vectors to the hyperplane is referred to as margin. There are multiple possibilities of placing the hyperplane in the space [20, p. 1506]. In order to obtain the best generalization capacity, the SVM places the hyperplane in a position, that maximizes the margins.

The example in figure 8 is for demonstrating the basic idea of SVMs. Real-world scenarios are often more complex. The populations can be partly overlapping and not linear segregable. Figure 9 exemplifies a population distribution that cannot be divided by a hyperplane.

In order to be able to separate data points by a hyperplane, nonlinear problems are conventionally transferred into a higher dimension. Figure 9 shows a hyperplane of a higher dimension reduced to the original dimension of the problem. Hyperplanes in higher dimensions can be approximated by a kernel-function. There are multiple common kernel functions [49], as Gaussian (used in figure 9), Poly-

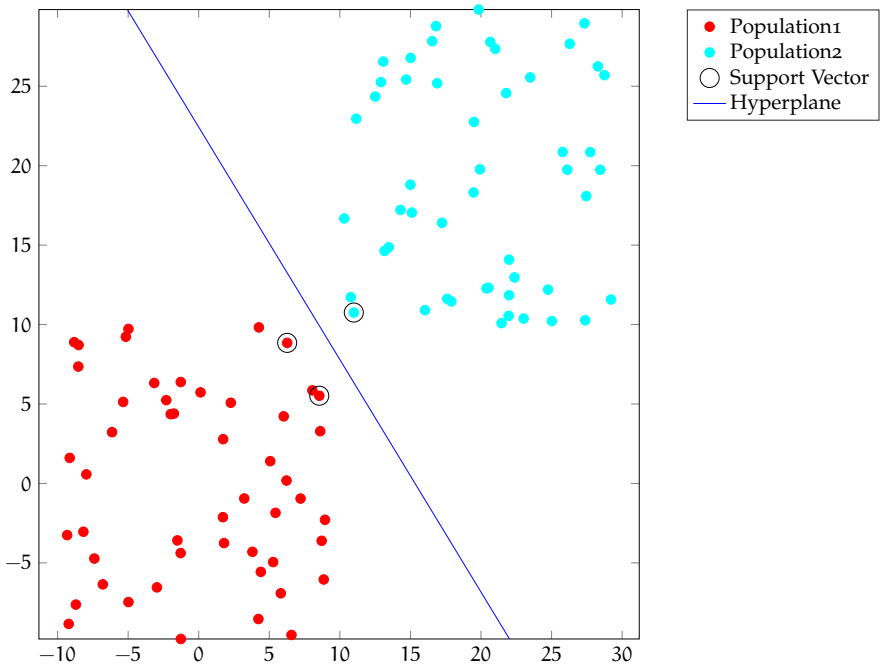


Figure 8: Example of a 2-D SVM

nomial, Laplacian and others. For specific problems, custom kernel functions can be defined. Further informations on the selection of kernel functions can be found in [50] and [49]. The shown examples are two dimensional. SVMs in general, are capable of handling high-dimensional data points as well.

If a new data point is presented to the SVM, the distance of this data point to the hyperplane is rated. Depending on the side of the hyperplane, the new data point is placed in, the SVM assigns the data point to a population. The distance of the data point to the hyperplane can be understood as measure of certainty of the decision made by the SVM.

2.5.2 *Gaussian Mixture Models*

A GMM is a parametric probability density function. It is compound of a weighted sum of Gaussian component densities [20, p. 827].

An univariate Gaussian distribution, also referred to as univariate normal distribution, is defined by:

$$\Pr(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-0.5(x-\mu)^2}{\sigma^2}\right], \tag{28}$$

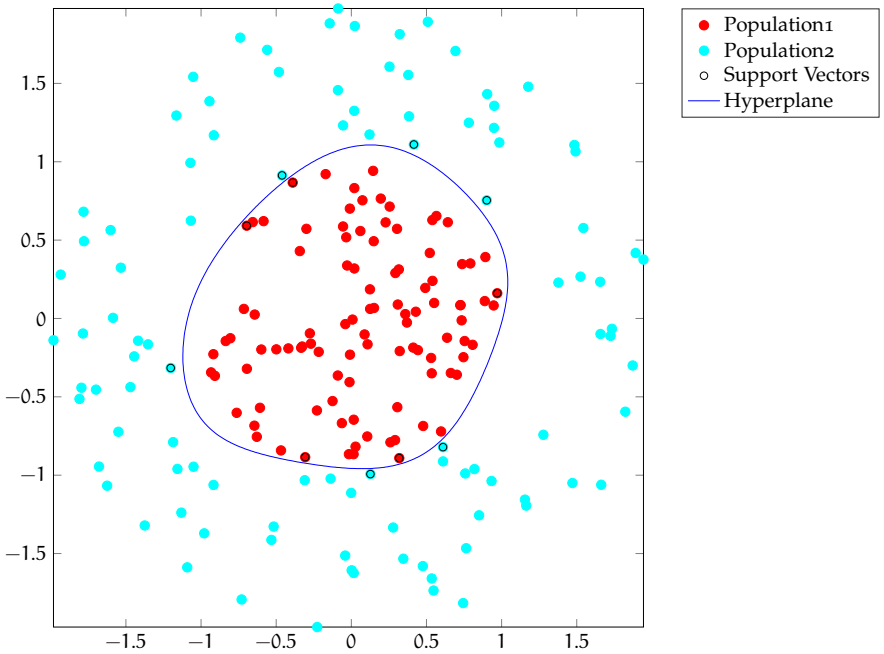


Figure 9: Example of a SVM with non-linear separable 2-D data

with a mean, μ, determining the position of the distribution, and a variance σ^2 determining the shape of the Gaussian bell [51, p. 40]. Figure 10a shows a univariate Gaussian distribution with $\mu = 0$ and $\sigma = 1$.

In order to represent multivariate Gaussian distributions, equation 28 has to be extended. The Gaussian distribution for D-dimensional variables is defined as

$$\Pr(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]. \tag{29}$$

As for the univariate distribution, the mean-vector, μ, determines the position of the Gaussian distribution. The shape of the Gaussian distribution is defined by the covariance-matrix Σ. The univariate distribution can be seen as a special case of the multivariate distribution with $D = 1$ [51, p. 41]. Figure 10b presents a multivariate Distribution for $D = 3$.

In a real-world scenario, most densities of observations can not be represented by a normal distribution. In order to model a more flexible distribution, the GMM is a weighted sum of M component Gaussian densities. The density of a GMM is given by:

$$\Pr(x|\lambda) = \sum_{i=1}^{M} w_i \cdot \Pr(x|\mu, \Sigma). \tag{30}$$

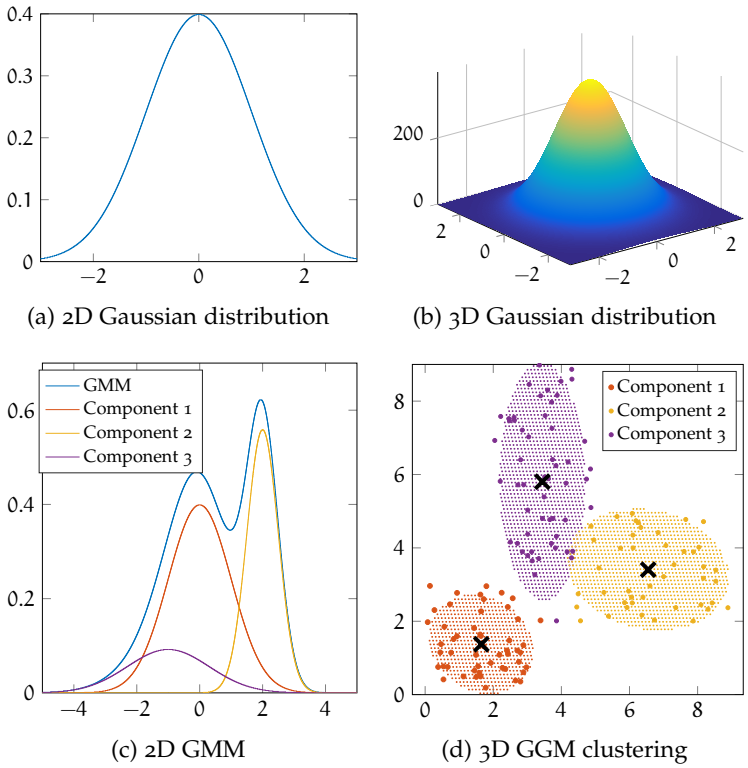
$Pr(x|\mu, \Sigma)$ is defined in equation 29. Vector w represents a weight for each component i of the GMM. λ contains a representation of the GMM parameters, which are denoted as:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, \quad i = 1, ..., M. \tag{31}$$

Figure 10c shows an example of a 2D-GMM with three components, the three weighted components are displayed as well. The parameters of the given example are set to:

$$\lambda = \left\{ \begin{array}{ccc} 1, & 0, & 1 \\ 0.7, & 2, & 0.5 \\ 0.3, & -1, & 1.3 \end{array} \right\}. \tag{32}$$

In order to represent a population, the parameters of the GMM have to be adapted with training vectors. There are multiple methods for estimating λ, common methods are Maximum Likelihood (ML), Expectation Maximization (EM) or Maximum A-Posteriori (MAP). As EM is utilized in this thesis, it will be focused in this chapter. Due to timing constraints, other estimation algorithms are excluded.



(a) 2D Gaussian distribution (b) 3D Gaussian distribution

(c) 2D GMM (d) 3D GGM clustering

Figure 10: Examples for Gaussian distributions and GMMs

The EM algorithm is an iterative process, that aims to increase the probability for a sample in the GMM in each iteration step. The process stops after a defined number of iterations or if the model reaches a convergence threshold.

The training process of the GMM is a probabilistic clustering process. The shape of the components is adapted to the distribution of the training samples. Figure 10d shows a result of a finished training process on a set of data samples. The different colours do not characterize different populations, but different components of the GMM and its affecting samples. The depicted example is trained on one population and can be used to determine the probability of a new data point belonging to the known population. Each component contributes to the overall probability depending on its weight, where the overall probability is normalized to 1.

For classification tasks, one GMM has to be trained for each hypothesis. If a new sample is presented to the trained GMMs, the LR of this sample fitting in the given distribution can be calculated utilising equation 30.

3

SPOOFING AND PRESENTATION ATTACK DETECTION

3.1 SUBVERSIVE USAGE OF BIOMETRIC SYSTEMS

Biometric systems are used for identification and verification purposes. In every verification or identification scenario, a subversive biometric capture subject is imaginable which aims to get verified as a different subject or does not want to be identified. Also, subversive users are possible, for instance, an administrator who manipulates the system, in order to allow access to not enrolled subjects to the system.

The aim of subversion attemps can be distinguished in: Attempts to get someone verified as an other subject, referred to as attacks on biometric systems in this thesis, and attempts not to be identified, referred to as concealment, which is in accordance to ISO/IEC 30107-3-CD2 [15]. As this thesis aims on PAD methods trying to gain access to the system, concealment will not be considered any further.

3.2 ATTACKS ON BIOMETRIC SYSTEMS

There are several possibilities to attack biometric systems. As displayed in figure 11, the attacks can be separated into attack points. Point 1 and 2 are direct attacks which are performed in front of or directly behind the sensor. The focus of this thesis is set on the first two points. As they are easy to perform and have a high success rate, they are considered to be more risky than attacks at the other points [14]. Point 3 to 9 are indirect attacks, as the attack is not on the input signal. But the system itself is influenced to change the output.

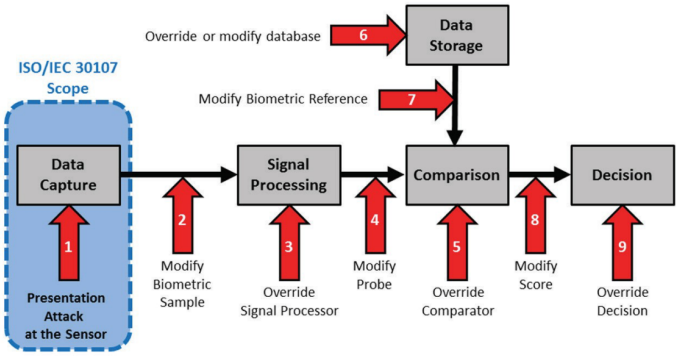


Figure 11: Possible attack points of a biometric system, according to ISO/IEC 30107-1 [52]

3.3 ATTACKS ON SPEAKER RECOGNITION SYSTEMS

In the field of speaker recognition, direct attacks are referred to as spoofing or Presentation Attacks (PAs). The recognition of such attacks is called PAD [52]. Direct attacks on speaker recognition systems can be divided at least into 6 attack types:

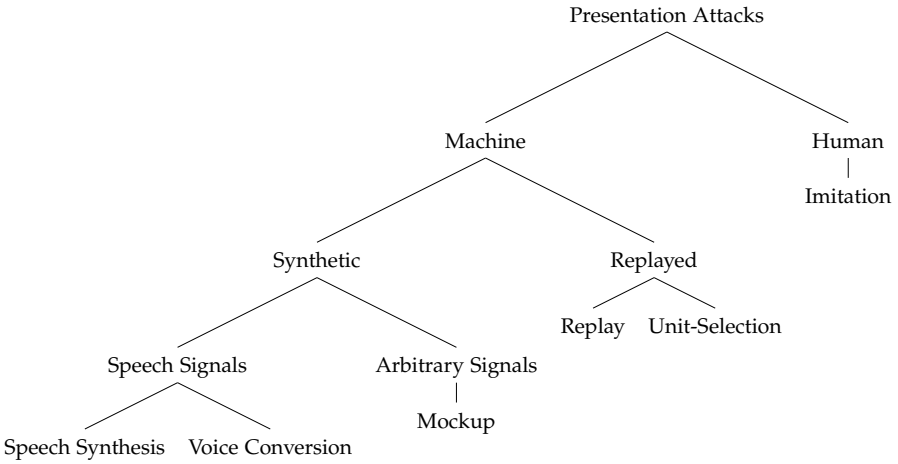


Figure 12: Structure of presentation Attacks on SIV systems

3.3.1 *Imitation*

Definition: Imitation or Mimicry is the attempt of an impostor to mimic a subject that is enrolled in the system, to get access to the system via the foreign account. There is a difference between skilled and unskilled imitators.

Attack: For this attack, no further technical equipment or algorithms are needed. The success of this attack strongly depends on the attacker and the attacked subject. The population of speakers can be divided into at least four subsets, referred to as sheep, goats, lambs and wolves. Sheep is the default speaker type, goats are difficult to recognize, lambs can easily be imitated, and wolves are successful in imitating other subjects. [53]. The voice of blood-relatives is more similar, therefore the imitation of a blood-relative should be treated separately.

Countermeasures: As imitation attacks can be fend with SIV systems with good baseline performance [14], they will not be discussed any further in this thesis. The performance of state-of-the-art SIVs is only slightly affected by imitation attacks [54].

3.3.2 *Mock-up Signal*

Definition: This is a rudimentary synthetic attack. The Impostor tries to generate a synthetic signal which aims to circumvent the classificator of the SIV.

Attack: A synthetic signal is generated by the attacker. As the signal does not need to sound like voice or conceptually provide quality standards, it can be generated with random noises or simple algorithms. The only requirement is that the system assesses the signal to match a mock-up probe to an enrolled reference. For text-dependant SIV systems, a combination of a mockup signal with other attack methods can be successful [55].

Countermeasures: The reviewed literature regarding mock-up signals [55], targets solely GMM-UBM systems, and provides no further countermeasures. However, by considering i-vector/PLDA systems as target, mock-up signals would need to bypass JFA, LDA, WCCN, and radial gaussianization in order to be capable of achieving a good enough score within the hidden PLDA subspace, which is assumed to be rather challenging and left for future research.

3.3.3 *Speech Synthesis*

Definition: The attacker creates a synthetic voice of the attacked identity. There are multiple methods for speech synthesis, some are described in section 2.3.

Attack: Speech synthesis attacks are usually two staged. The first stage generates the linguistic and phonetic elements which will be synthesised, e.g. parsing a text and partitioning the words into phonemes. The second stage is generating the acoustic signal that can match with the linguistic and phonetic elements of the victim [14].

Countermeasures: The attack potential of synthesis attacks against SIV systems is quite high [12, 56]. As most synthesis algorithms produce artefacts in the synthesised sample, many PAD systems for synthesis attacks aim on detecting these artefacts [14]. HMM based synthetic speech can be discriminated from human speech by estimating the intra-frame difference of the samples [57]. Another approach is the usage of MFCCs. Most synthetic voices are smoothed afterwards, so the higher order cepstral coefficients differ from those of human speech [58]. These methods are algorithm dependant. A more algorithm independent approach is the evaluation of the differences of samples generated by vocoders and natural speech. For example the spoof detection with phase-based features is highly successful

against synthesis [4, 59, 60]. As the smoothing of the signal affects the prosodic characteristic, the f0 patterns in synthetic speech are smoother than in human speech [14].

3.3.4 *Voice Conversion*

Definition: For voice conversion attacks, the speech signal of the impostor is changed in a way that it becomes more similar to the voice signal of the target subject [14].

Attack: The proceeding is comparable to speech synthesis. The input is not a text but natural speech. Typically the timber or prosodic characteristics are adapted, like fundamental frequency or duration [14].

Countermeasures: Attacks with voice conversion are in general very efficient. For detection the same countermeasures as for speech synthesis attacks can be used [14].

3.3.5 *Replay Attacks*

Definition: Replay attacks are the most basic kind of attacks on SIV systems. The voice of a subject is recorded and later played to the SIV system [14].

Attack: No further knowledge of speaker recognition or signal processing is needed for this attack. As technical equipment a microphone is required for the recording of a replay sample and a speaker replaying the recorded speech signal in order to attack a SIV system [14].

Countermeasures: Because of its simplicity, independence of further knowledge of speech processing and efficiency against modern SIVs systems, replay attacks are a present threat [12]. Due to the spreading of mobile phones, the risk is increasing [14]. Despite the simplicity of replay attacks, in current research published there are only a few studies [14]. A simple way of avoiding this kind of attack is using challenge response, where the SIV asks for a specific e.g., randomized sentence. For a successfull attack the attacker has to know the right phrase.

A more attack specific detection is based on analysing the SNR. If the voice signal of a person is captured, the noise of the environment and the microphone is added. In the case of a replay attack the noise of environment and microphone during the capture process are added as well. This difference in background noise can be detected.

The second approach focuses on spectrum and modulation. Recording and replaying flattens the spectrum and reduces the modulation. This difference can be detected, for instance, by a SVM [14, 61].

3.3.6 *Unit Selection*

Definition: A more sophisticated kind of replay attacks is unit-selection. Speech samples of the attacked subject are recorded and divided into parts, called units. These units are later replayed in different sequence to the SIV system. In some definitions, unit-selection is classified as concatenative speech synthesis, as unit-selection is capable of TTS synthesis [29]. However, in this work, unit-selection is rather classified as a kind of replay attack. The replayed units are only altered in its order, where the classic replay attack can be seen as a special type of unit-selection with one unit.

Attack: The recorded speech is divided into units. Units can be words, diphones, phones, or even smaller parts. In order to attack the system, units are concatinated and replayed. With this technique it is possible to overcome SIVs-Systems using challenge response. A more detailed description of the process of unit-selection can be found under section 2.3.4.

Countermeasures: Unit-selection is an effective attack against modern SIVs-Systems and PAD-Algorithms [11]. As unit-selection can be considered as short-time replay attacks, the attack specific countermeasures introduced for replay attacks can be used. The modelling of prosody is still a difficult task for unit-selection algorithms. At the concatenation point of two units, the formant f0 tends to jump. The statistical behaviour of the formants can be used to detect unit-selection attacks [62, 63]. An additional method for unit-selection detection, based on analysis of the frequencies, will be processed in this thesis.

3.4 PRESENTATION ATTACK DETECTION FOR SIVS

As shown in section 3.3, there is a large range of possible attacks for speaker recognition systems. The effectiveness of countermeasures depends on the attack they are designed to encounter for. Challenge response for example, is useful against replay attacks, but cannot be used against attacks synthesizing sentences. In addition, it is not applicable for forensic scenarios. To a certain extend, the introduced countermeasures are useful to detect multiple attacks. Prosodic features, for example, can be utilized for detecting synthesis attacks as well as unit-selection attacks [14].

Independent from the structure introduced in figure 12, the attacks can be divided into two groups: text-dependent and text-independent attacks. Replay attacks for example are strongly text-dependent, as only the recorded sentence can be played to the SIV system. If the system requires a sentence not available, text-independent attacks are more likely to succeed. Most studies about PAD examine the effect of a countermeasure to a specific attack. This assumes that the attacked system knows the attacker and can prepare for this specific attack. It would be of interest, which countermeasures are able to detect a larger set of attacks. PAD algorithms for speech synthesis attacks should be able to detect voice conversion attacks as well, as both attacks produce similar artefacts while producing the signal. The impact of the combination of multiple countermeasures needs further research.

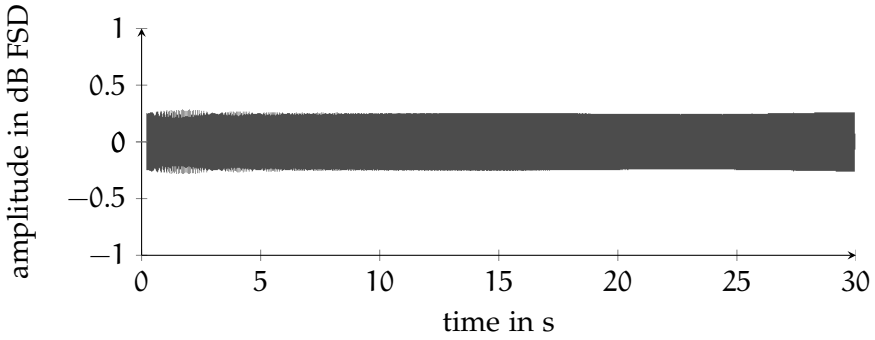
Imitation and Mock-up attacks are not considered as threat, as they are reliable detected by state-of-the-art PLDA SIV systems [54]. Current research provides PAD algorithms which are able to detect voice conversion and synthesis attacks at EERs of almost 0% [4, 60]. Thus, this section focuses on the creation of replay, also unit-selection attacks, and the state-of-the-art for detecting unit-selection attacks, as current research lacks of PAD subsystems for these specific attacks.

3.5 CREATION OF REPLAY ATTACKS

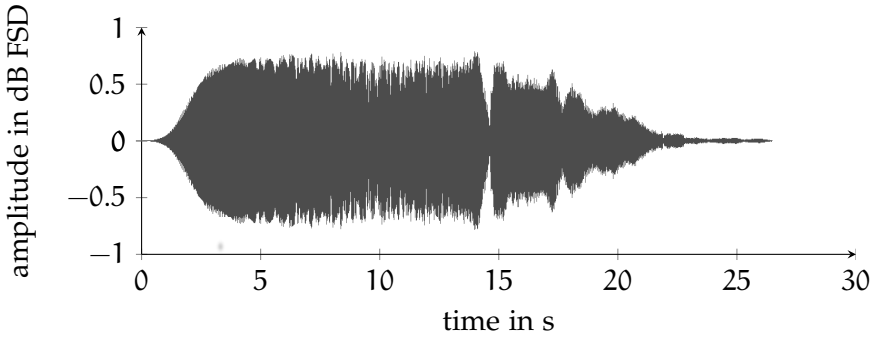
In the context of this thesis, a database of replay attacks is generated. For this task, a speaker and a laptop are utilized. The speaker is used for a clean replay, so a high quality Near Field Monitor (NFM) is employed. It provides a linear frequency response, so the corruption of the replayed signal is minimized. As NFMs are expensive and less mobile, a MacBook Pro is used to generate a more realistic attack.

The database should reflect attacks on mobile authentication devices, as mobile phones. For the purpose of covering a great variety of current smartphone microphone technology, four smartphones from low-end to high-end solutions were examined, in particular: Samsung Galaxy Note 4, HTC One, Motorola Moto G, and LG G2.

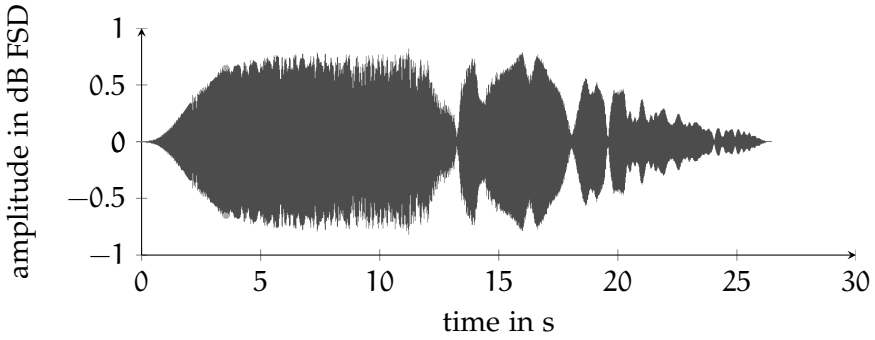
The NFM has separated membranes for high and low frequencies. As result the position of the microphone in front of the NFM affects the transmission of speech signal. Figure 13 shows the waveform representations of a sweep signal from 20 Hz to 20 kHz.



(a) Clean sinus sweep with constant amplitude over all frequencies



(b) Sweep replayed by the NFM and recorded by the Samsung Galaxy Note 4. Heigth 0 cm, distance 10 cm



(c) Sweep replayed by the NFM and recorded by the Samsung Galaxy Note 4. Heigth 16 cm, distance 10 cm

Figure 13: Waveform representation of a sweep signal from 20 Hz to 20 kHz

The original, artificial generated signal, figure 13a, has a constant amplitude. The non-linear microphone of the mobile phone causes a distortion of the signal. High and low frequencies are recorded less accurate than the midrange. The indentations in the signal are caused by interferences due to reflections.

If the mobile phone is put on the table, it is in front of the woofer (figure 13b). The amplitude for low frequencies ascends faster. If the mobile phone is placed higher, it is closer to the tweeter (figure 13c). In consequence, high frequencies have higher amplitudes compared to figure 13b. The interferences are affected by multiple factors, for example the surrounding, the distance, and the height of the microphone. In order to find the optimum set-up for NFM and microphone, different distances were tested and compared.

For the comparison of the captured signals Dynamic Time Warping (DTW) is employed. DTW is a common algorithm for estimating the nonlinear time synchronization between two signals [20, p. 786]. Even signals with different length can be compared. The DTW algorithm is two-staged. First, the spectrograms of the probe audio signals are calculated. For each window, the costs for morphing the signal to each window of the reference signal are calculated.

Experimental set-up: The mobile phones are placed with the microphone towards the NFM. In order to determine the height and distance between the NFM and the microphone causing the lowest costs when matching reference and probe, different scenarios are examined. The height is varied from 0 to 20 cm in 4 cm steps. The distance is varying between 10 and 50 cm in 10 cm steps.

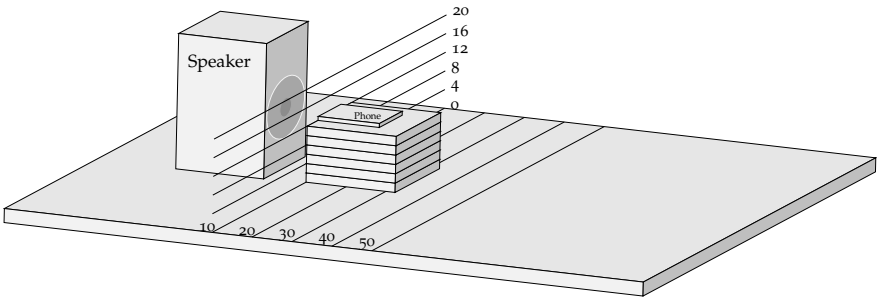


Figure 14: Different positions for the placement of the microphone

For testing, a sweep from 20 Hz to 20 kHz, displayed in figure 13a, and a sample of human speech are replayed by the NFM and the MacBook. The signal is recorded by all four mobile phones for every position shown in figure 14. In contrast to the speech signal, the sweep signal covers all frequencies with the same amplitude. As the speaker of the MacBook is not designed for linear frequency response,

it is likely the comparison score of the MacBook-replays differs a lot from the NFM-replay.

Experiments: It can be assumed, that replaying and recording leads to a distortion of the audio signal. In order to determine a quantitative value for distortion, each recorded signal is compared to its reference signal via DTW. Figure 16 depicts the heat-map for the comparison of a recorded speech signal and its origin. Light fields display high costs, dark fields have lower costs.

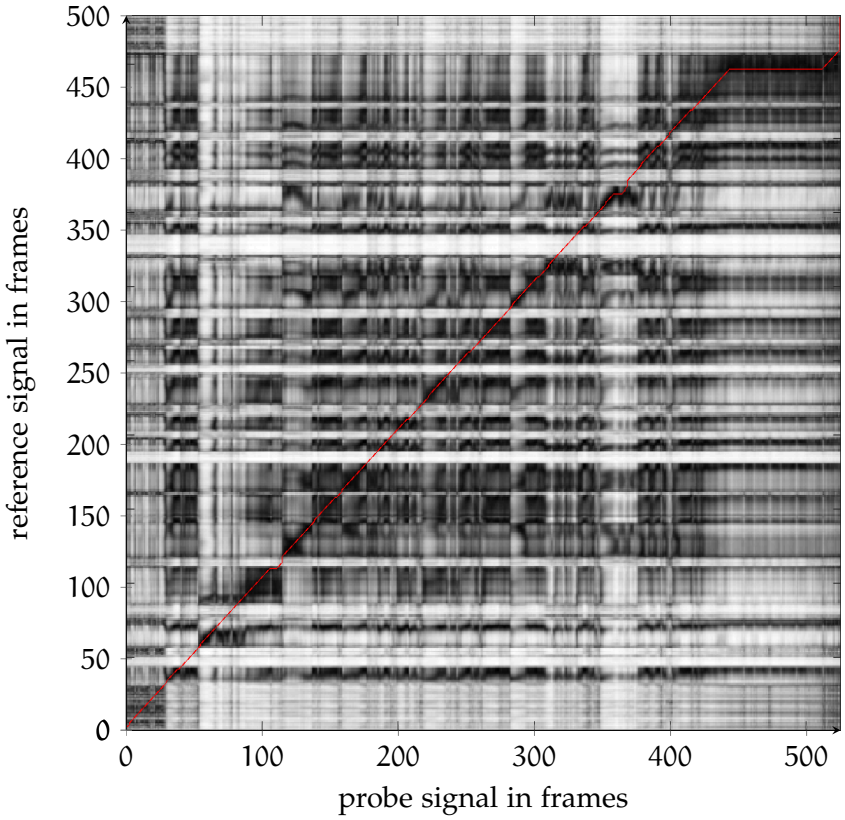


Figure 15: Similarity matrix of a probe speech signal and the reference

The second step of the DTW algorithm is finding the path with the lowest costs. The shortest path in the given example is represented as a red line. Figure 17 shows the similarity matrix for a recorded sweep signal.

In contrast to the similarity matrix of the speech signal, the comparison of the sweep signal shows a clearly visible path amongst the low costs. This is caused by the steadily increasing frequency of the signal, which enables a more precise assignment as the windows of the spectrogram are more distinct.

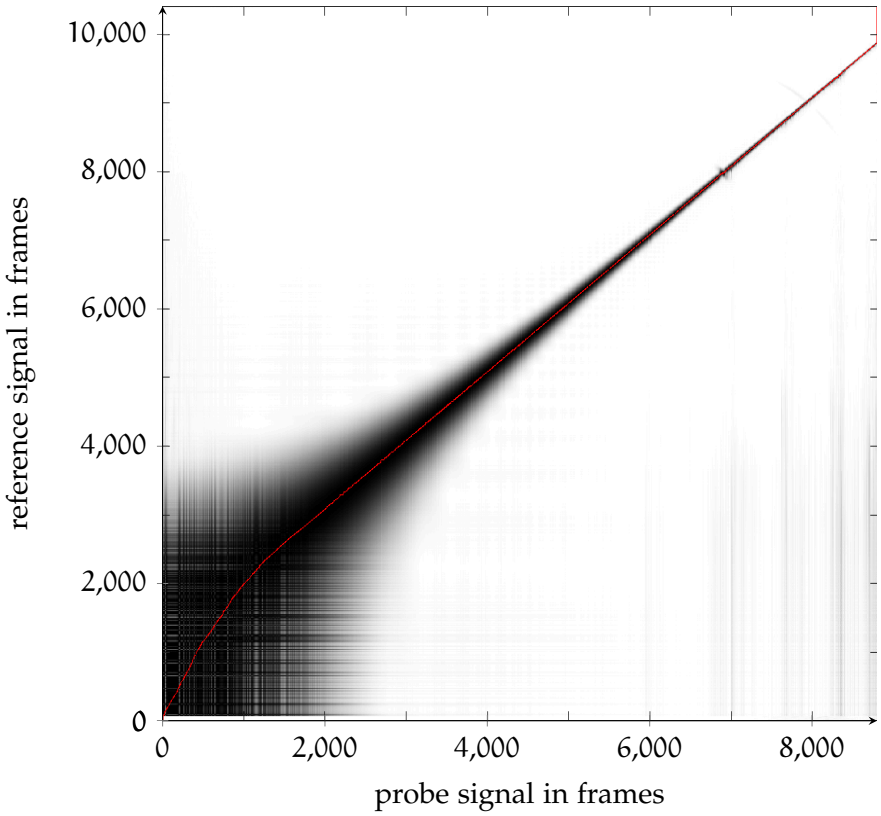


Figure 16: Similarity matrix of a recorded sweep and the original signal

In order to provide a scalar measurement, a metric for conveying the path into a single value is needed. A simple, yet common, method is the sum over all costs on the path. The resulting DTW-score displays a measurement for the difference between the original signal and the compared one.

Each recorded signal is compared with the origin, resulting in a score for each possible combination of NFM, microphone, height, and distance for human speech and sweep signal.

Evaluation: Figure 17 visualizes the influence of distance and height in combination with the different speakers and microphones. The difference between samples, replayed by the laptop and the NFM are enormous. The highest similarity score of recorded laptop-replays is lower than the lowest score of the recorded NFM-replays. This behaviour is expected, as the NFM provides a linear frequency response for the replayed frequencies. The laptop in contrast is not designed for high quality audio replays. In particular, the lower frequencies cannot be replayed properly.

The next factor, influencing the similarity score, is the microphone. The microphones of the high quality mobile phones are not as good

as the microphones of the cheaper products. Another factor to carry weight is the height of the microphone. In general, positions above 16 cm cause better comparison scores, than lower positions. A distance of to 10 cm to 20 cm shows the best results. The best overall score for sweep signals is achieved with the motorola placed at a height of 16 cm and a distance of 10 cm, recording a sweep replayed by the NFM.

In a real-world scenario, a replay attack does not record a sweep, but human speech. Figure 18 shows the similarity scores achieved, when recording a human sample. As this chapter aims for high-quality replay attacks, the results for the NFM are examined.

The influence of height, distance and microphones is vanishing. In contrast to the sweep signal, the expensive mobile phones are good for recording samples replayed by the NFM. This behaviour can be caused by the frequency response and filters of the higher class mobile phones. As a mobile phone is designed for human speech, frequencies below or above may distort the communication. The effect of distance and height is not as conspicuous as for the recording of the sweep signal. Still, nearer distances and a height between 8 and 16 cm produce best results. Again, the overall best result is achieved by the motorola, placed in 12 cm height and a distance of 10 cm.

Conclusion: The recording of sweeps or human speech seriously influences the similarity of the recording to the original sample. Microphones of upper class mobile phones are far better in recording human speech samples than sweeps.

The purpose of this section is to indicate a reasonable positioning of the microphone for the generation of replay attacks. The highest similarity score is achieved for a height of 12 cm and a distance of 10 cm.

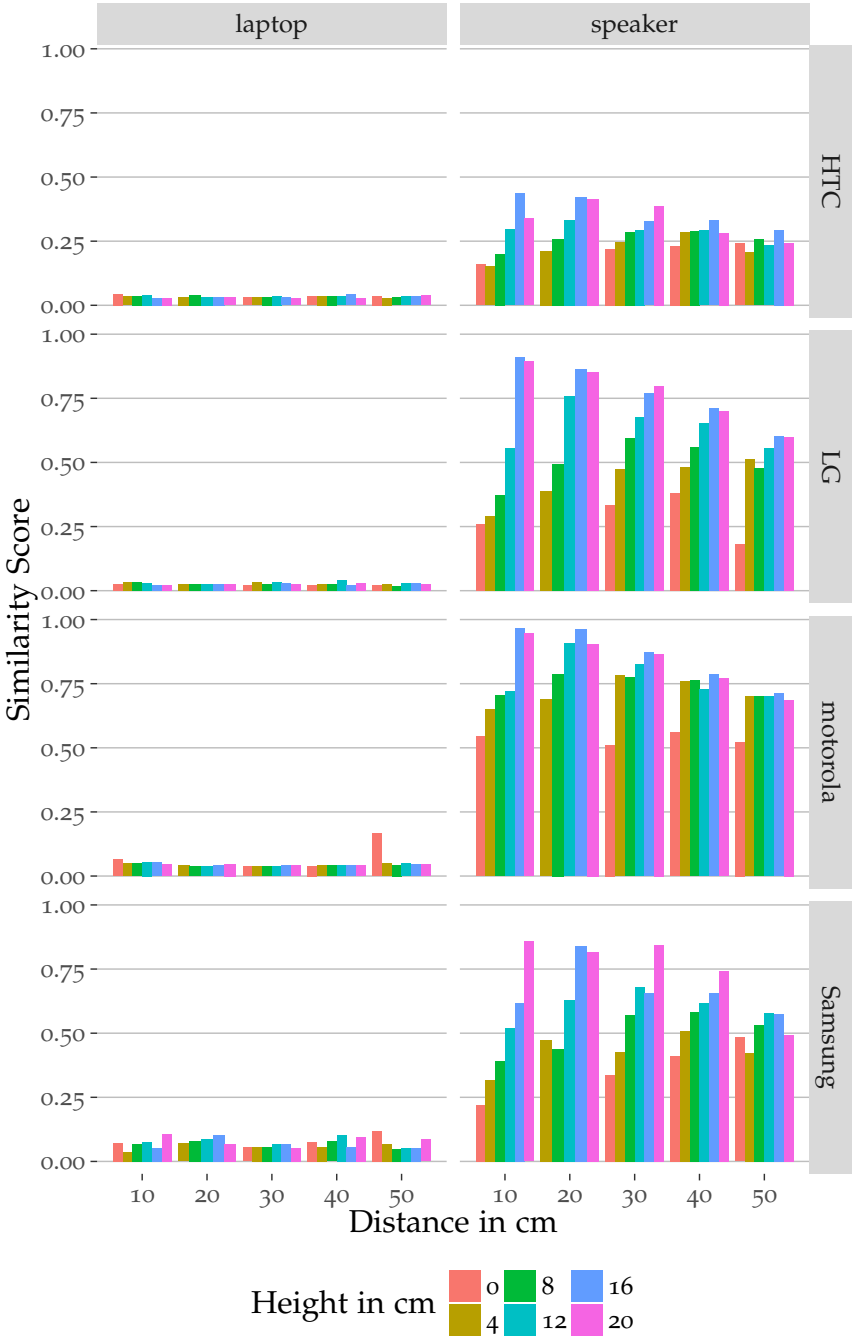


Figure 17: Relation between distance and similarity score for a sweep signal

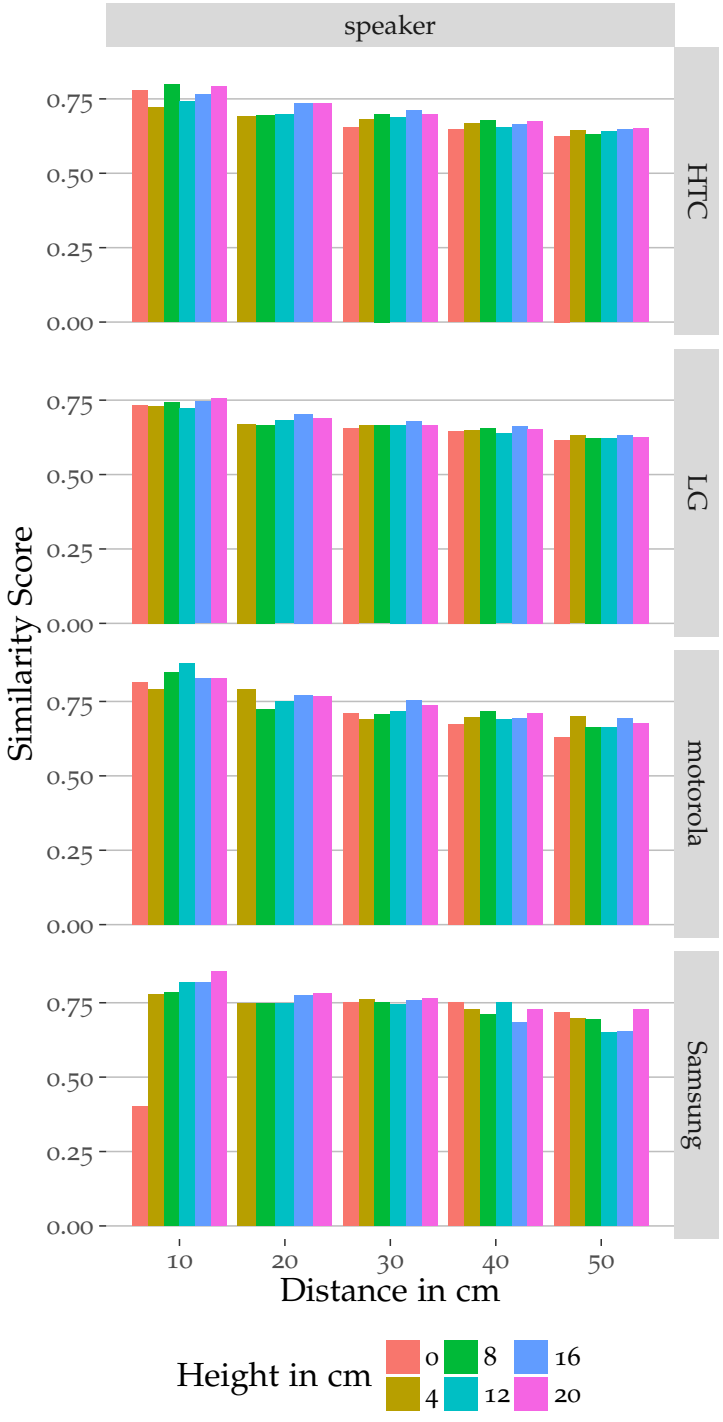


Figure 18: Relation between distance and similary score for human speech

3.6 CREATION OF UNIT-SELECTION ATTACKS

In this thesis, focus is placed on PAD for unit-selection attacks, the execution of unit-selection attacks is described in detail in this section. In general, the production of unit-selection attacks requires at least three steps. The acquisition of transcribed speech data of the attacked person, the creation of a unit-selection voice, and finally the generation of unit-selection samples. The emphasis of this section is put on the technical part of unit-selection attack creation, thus a transcribed database with clean speech data of the attacked subject is assumed.

3.6.1 *Creation of Unit-Selection Voices*

The creation process of a unit-selection voice is introduced in accordance to the unit-selection tutorial of MaryTTS [64]. The creation of a unit-selection voice consist of at least eight steps. A unit-selection voice consists of a set of units and Classification And Regression Tree (CART) trees for selecting the convenient unit, the eight steps are the following:

Acoustic Data: This is the first analysis of the raw, unprocessed audio data. The MFCCs of the samples are calculated and the pitch, the f0-parameter, is determined.

Automatic Labeling: The transcription files for the speech signal have to be processed. Phonemes can be pronounced different, the possible pronunciations are called allophones. The different allophones are pre-calculated from the transcription files. In a second step, phonetic labels are extracted from the transcription files.

Label-transcription Alignment: The previously estimated phonetic labels are aligned with the transcription files. Also the allophones are aligned with the transcription. Each allophone is now assigned to a phonetic label.

Feature Extraction: Features are extracted from the allophones. Features can be the previous phoneme, the next phoneme or the phone the allophone belongs to. There are numerous more features that can be extracted from the allophone.

Verify Alignment: The features generated in the previous step are aligned with the labels generated by the automatic labelling process. After the alignment process, each phonetic label is assigned to a feature vector.

Basic Data Files: The raw audio data is divided into unit files using the pre-calculated f0-parameter.

Acoustic Model: The features of the units are mapped to the according audio unit created in the previous step. The costs for the mapping of each unit to the next is calculated. In order to be able to find a proper unit to a phoneme, two trees are trained. One for the duration of each unit, a second for the f0-parameter.

Unit-Selection Files: In the last step, the target costs for each unit are determined. A tree for the target costs is trained. The creation of the unit-selection voice is finished.

The finished unit-selection voice contains a list of units and four CART trees. The trees for f0-parameters and duration can be used to estimate the characteristics for a unit, the trees with target and join costs can be used for finding a suitable unit.

3.6.2 *Creation of Unit-Selection Attack samples*

The generated unit-selection voice can be used for TTS synthesis with the voice of the attacked subject. If a new text is provided to the unit-selection voice, it is divided into phonemes. The duration and f0-parameter tree of the voice are used to estimate a suitable unit for each phoneme. Utilizing the trees for join and target costs, the best fitting unit is selected.

3.6.3 *Quality of Unit-Selection Voices*

The success of the creation of a unit-selection voices highly depends on the samples used for training. It is obvious, that a correct labelling of the samples is vital for the creation of an accurate unit-selection voice. In order to improve the labelling process, a preceding normalization and VAD on the samples is recommended [65].

One possibility to determine the quality of a unit-selectionn voice is the phonetic coverage [66]. A good voice should cover every diphone, a perfect voice should contain multiple prosodic realisations for one diphone. The coverage of diphones depends on the overall length of the used samples [65]. In this thesis, the quality of a unit-selection voice is evaluated by the efficiency of the attack against a SIV system.

3.6.4 *State-of-the-art Algorithms*

In the context of ASVspoof 2015, new PAD algorithms against synthesis are introduced [4, 5, 6, 11, 59, 60, 67, 68, 69, 70, 71, 72, 73]. The

focus of the challenge is set to synthesis attacks. Most detection algorithms utilized machine learning algorithms evaluating phase-based features combined with well established features like MFCCs. As the main focus of the ASVspoof is set on synthesis and voice conversion, most submitted algorithms did not perform well in detecting unit-selection attacks. The algorithm proposed in [6] achieved an EER of 8.5% when detecting unit-selection attacks.

The approach, introduced in [6], utilizes new defined Cochlear Filter Cepstral Coefficient Instantaneous Frequency (CFCCIF) features. The CFCCIF is a combination of Cochlear Filter Cepstral Coefficients (CFCCs) and Instantaneous Frequency (IF).

CFCC: First introduced in [74], the CFCC is calculated by utilizing an Audiotory Transform (AT), followed by a filter bank and a Discrete Cosine Transformation (DCT). The AT itself is a function emulating the filter function of the cochlear [75]. In the proposed implementation, the filter utilizes the wavelet transformation, equation 25, with a specialized wavelet, defined as [74]:

$$\begin{aligned}\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\left(\frac{t-b}{a}\right)^{\alpha} exp\left[-2\pi f_l \beta\left(\frac{t-b}{a}\right)\right] \\ \cdot cos\left[2\pi f_L\left(\frac{t-b}{a}+\theta\right)\right] u(t-b),\end{aligned} \tag{33}$$

whereas α and β define the shape and width of the filter, θ is selected to satisfy equation 26.

In a next step, the filtered signal is mapped by a filter bank, as the basilar membrane of the human ear processes acoustic signals. The resulting signal is decorrelated applying a DCT [6].

IF: The IF is defined as the derivation of the unwrapped phase [6]. The analytic signal, $s_a(t)$ of $s(t)$ is defined as:

$$s_a(t) = s(t) + j s_h(t), \tag{34}$$

whereas $s_h(t)$ is the Hilbert transform of $s(t)$ [6]. The unwrapped instantaneous phase $\phi(t)$, is defined as the argument of the analytic representation $s_a(t)$:

$$\phi(t) = tan^{-1}\left(\frac{s_h(t)}{s(t)}\right). \tag{35}$$

As the IF is defined as the derivation of the unwrapped phase $\phi(t)$ it can be calculated as:

$$IF = \frac{d\phi(t)}{dt}. \tag{36}$$

The CFCC itself does not consider phase informations. In order to create a feature that includes phase information, the calculated IF is inserted after the filter bank operation.

For the ASVspoof 2015 the CFCCIF feature is combined with MFCC features. For classifying, a GMM with 128 components is trained. With an EER of 1.2% the proposed algorithm performs well on the ASVspoof data set. The EER on the unit-selection attacks of the data set was 8.5% [6].

The ASVspoof-corpus is a first attempt in speaker recognition to reach a standard PAD evaluation protocol.

3.7 STANDARDS AND PROTOCOLS

In the current research for attacks and countermeasures, a universal standard for test protocols and databases is missing. The outcomes of different studies are hard or impossible to compare. Furthermore in most cases the datasets are to small to gain reliable and reproducible results [14].First steps for standardizing databases and protocols are the ASVspoof-corpus [68], the SAS-corpus [76] and ISO standards like ISO/IEC 30107-1, [52], and ISO/IEC 30107-3-CD2, [15]. An open source implementation for a baseline PLDA comparator is provided by the Voice Biometry Standardization Initiative[1].

3.7.1 *ASVspoof*

The ASVspoof database consists of 16 651 speech samples of 106 human speakers. 246 500 attacks have been generated by voice conversion, speech synthesis and replay attacks. The partitioning of the database is according to table 1. The database is a subset of the SAS-Corpus [76]. Ten different algorithms have been used for generating the attacks.

Table 1: Partitioning of the ASVspoof database

Subset	Bona Fide	Attack	Male	Female
Training	3 750	12 625	10	15
Development	3 497	49 875	15	20
Evaluation	9 404	184 000	20	26

1 http://voicebiometry.org

The ASVspoofing-challenge[2] was held in context of the Interspeech 2015[3]. Aim of the challenge is the possibility for researchers to submit and benchmark PAD algorithms. In order to allow participation for researchers, which have no previous knowledge in speech processing, only the accuracy of the PAD algorithm is assessed [68]. The interaction between PAD algorithm and SIV algorithm is excluded.

PAD algorithms, which estimate the phase and its shift, attract attention as they outperformed well on the given database [4, 5, 60, 71]. As the used vocoder does not reconstruct the phase shift, when synthesizing the signal, a coherence between the success and the vocoder used for generating the attacks is possible.

Thus, further investigations are needed in order to evaluate the reliability of the algorithms presented, if during creation of the attacks phase shift is considered.

3.7.2 *Voice Biometry Standardization Initiative*

The success of attacks depends on the implementation of the SIV system. In order to achieve comparable results, a standardized baseline-system is necessary. One attempt for the standardization of SIV systems is the voice biometry standard [77] of the Voice Biometry Standardization Initiative.

A software package with a standard conform implementation is free available[4]. The package is written in python and contains a script for extracting i-vectors from audio files, and a second for the PLDA score calculation.

3.7.3 *Metrics for PAD*

In order to obtain comparable and reproducible results from algorithm tests the use of standardized metrics is essential. The metrics common for speaker recognition, like minDCF, require the definition of an application specific prior. As the evaluation of PAD algorithms should be application independent, more universal metrics are required. A standard, ISO/IEC-30107-3 (Biometric presentation attack detection – Part 3: Testing and reporting), for PAD is in progress [15]. The two relevant metrics for the thesis are those for a subsystem performance evaluation:

2 http://www.spoofingchallenge.org/
3 http://interspeech2015.org/
4 http://voicebiometry.org/download/vbs_demo.tgz

APCER: Proportion of presentation attacks incorrectly classified as bona fide presentation at the PAD subsystem in a specific scenario. The APCER shall be calculated as:

$$\text{APCER} = \frac{1}{N}\sum_{i=1}^{N}(1 - \text{Res}_i), \tag{37}$$

where N represents the number of attack presentations. Res_i takes value 1 if the ith presentation is classified as an attack presentation, and value 0 if classified as a bona fide presentation. The APCER has to be calculated seperately for each Presentation Attack Instrument (PAI).

BPCER: Proportion of bona fide presentation incorrectly classified as presentation attacks at the PAD subsystem in a specific scenario. The BPCER shall be calculated as follows:

$$\text{BPCER} = \frac{\sum_{i=1}^{N_{BF}} \text{Res}_i}{N_{BF}}, \tag{38}$$

whereas N_{BF} represents the number of bona fide presentation. Res_i is defined as for the APCER.

APMR: In a full-system evaluation of a verification system, the proportion of presentation attacks in which the target reference is matched.

Attack Presentation Identification Rate (APIR): In a full-system evaluation of an identification system, proportion of presentation attacks in which the targeted enrolment is among the identifiers returned or, depending on intended use case, at least one identifier is returned by the system.

Further metrics are defined in the standard but not employed in this thesis:

Attack Presentation Non-Response Rate (APNRR): Proportion of presentation attacks that cause no response at the PAD subsystem or data capture subsystem responding.

Bona fide Presentation Non-Response Rate (BPNRR): Proportion of bona fide presentation that cause no response at the PAD subsystem or data capture subsystem.

PAD Subsystem Processing Duration (PS-PD): Milliseconds required for the PAD subsystem to classify PAD data.

Part II

DETECTION OF REPLAY, ATTACKS, AND ARTIFICIAL SPEECH

4

UNIT-SELECTION ATTACKS AND COUNTERMEASURES

As mentioned in section 3.3.6, unit-selection attacks pose a major threat to SIV systems. Figure 19 illustrates the baseline performance of the voice biometry standard algorithm, introduced in section 3.7.2, against a subset of the Open Speech Data Corpus for German [78]. The set-up achieves an EER of 10.3%. The measurement assumes a bona fide scenario, so no subject is intentionally attacked.

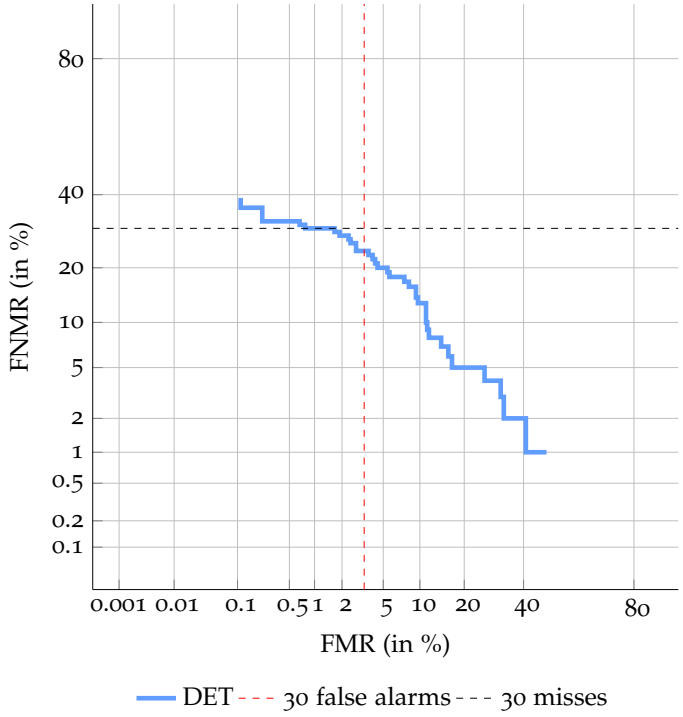


Figure 19: Baseline performance of voicebiometry.org algorithm

In addition to the baseline performance, the PAD performance of the voice biometric standard algorithm against unit-selection attacks is tested. The impostor comparisons for each subject are replaced with unit-selection attacks for the specific subject. A PAD resistant SIV system would not be influenced by the attacks, the EER would be equal or better compared to the bona fide use case. Figure 20 visualizes the PAD performance of the voice biometry standard algorithm.

The need for unit-selection detection algorithms is obvious. The value of the APMR never drops below 20%, the EER of the system against unit-selection attacks is 40.7%. As depicted in the PDF, figure 21,

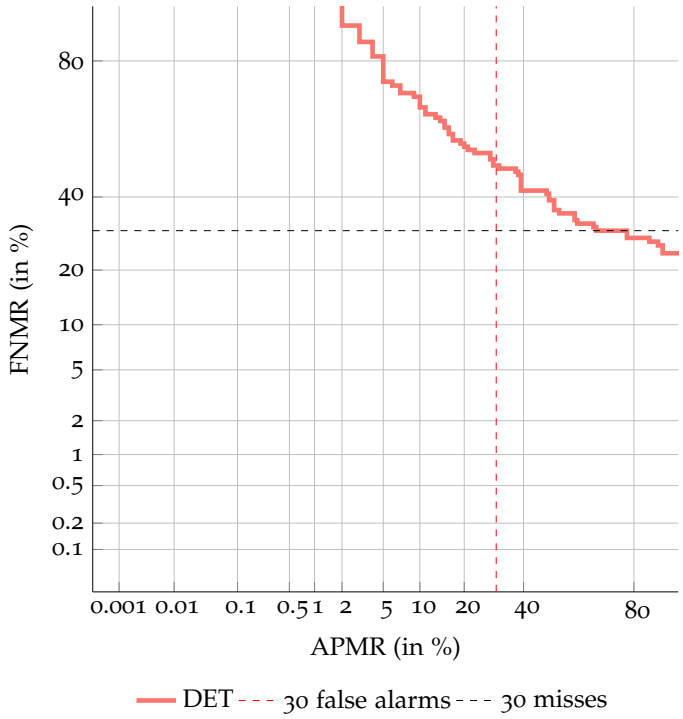


Figure 20: PAD capability of voicebiometry.org algorithm

the state-of-the-art PLDA comparator is not capable in distinguishing unit-selection attacks from bona fide speech samples, attack samples are even more likely to be accepted than genuine samples. Thus, PAD subsystems for unit-selection are strongly motivated.

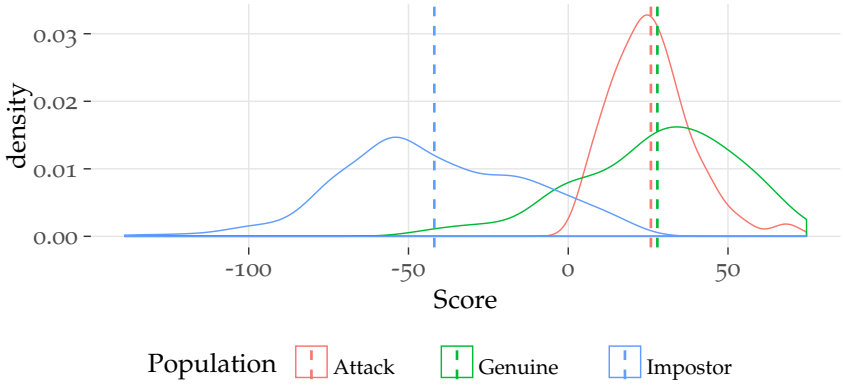


Figure 21: PDF of voicebiometry.org on unit-selection attacks

In order to develop a PAD system for unit-selection attacks, multiple approaches are examined in this thesis. State-of-the-art PAD systems utilize features imitating the human perception of speech [6]. The countermeasures introduced in this thesis are motivated by frequency analysis of an unfiltered signal. A basic algorithm generates

the sum over the differences of the analysed frequencies. The PAD performance of derived features can be optimized by machine learning techniques. The performance of the algorithms is evaluated with two independent data sets.

4.1 DETECTION ALGORITHMS

As shown in section 3.3.6, there are multiple possibilities for detecting unit-selection attacks. Figure 22 depicts a classification of different detection methods. In this thesis, the focus is put on frequency analysis.

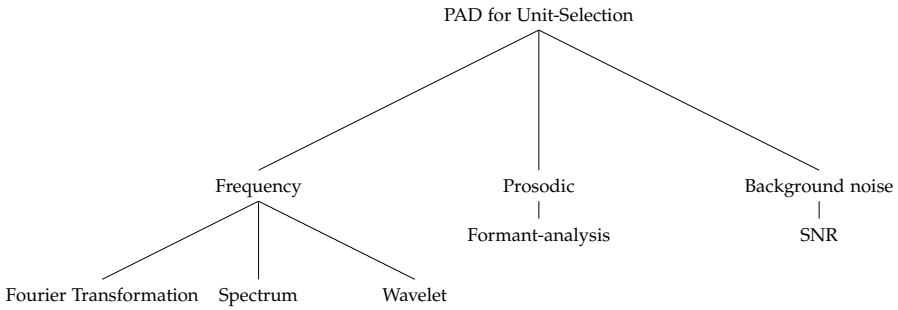


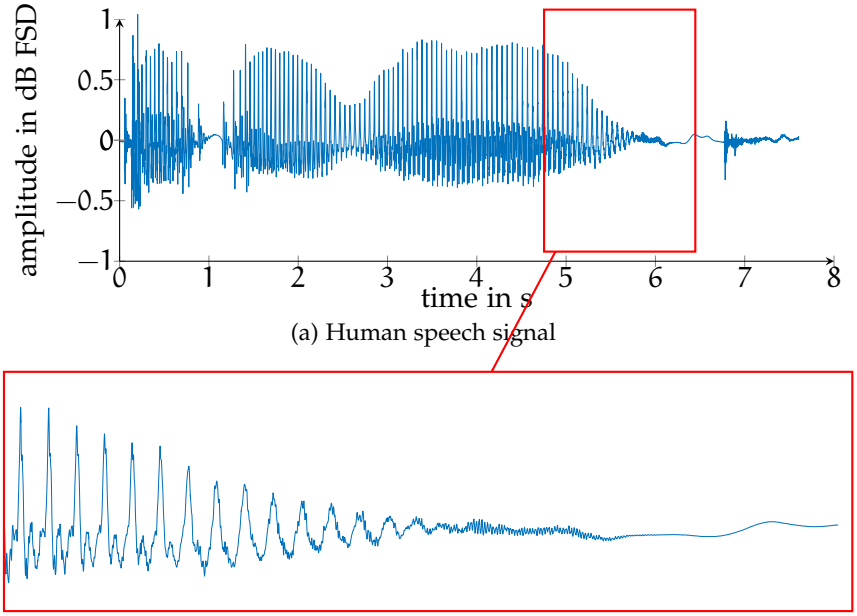
Figure 22: PAD for unit-selection attacks

The state-of-the-art algorithm introduced in section 3.6.4 utilizes an adapted CFCC in combination with MFCCs as feature vectors. Both features aim to simulate the perception of the human ear. The features introduced in this section do not utilize any filter banks, thus frequencies not perceived by the human ear are considered as well.

4.1.1 *Fourier-based Detection*

Speech is a concatenation of phonemes, the point of concatenation will be referred to as transition. In human speech, the phonemes are smoothly transferred into each other. The continuous transition of a human speech signal is displayed in figure 23. Audio-signals which are compound of multiple voice fragments and not smoothed afterwards show more abrupt changes of the frequency in the signal, as displayed in figure 25.

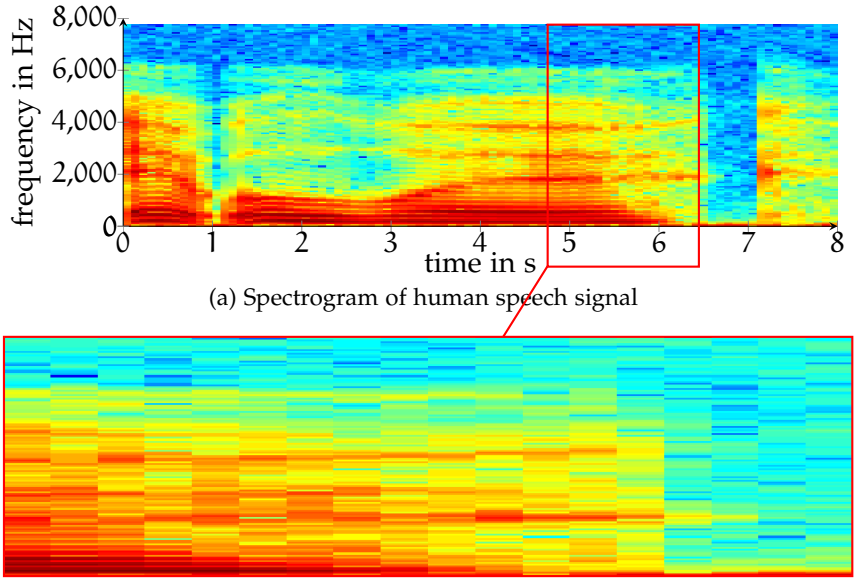
These abrupt changes are reflected in the frequency domain of the signal. The smooth transitions in human speech can be represented in the frequency domain by lower frequencies, whereas the transformation of the abrupt concatenated signal requires much higher frequencies. The spectrogram of human speech sample, figure 24, shows no frequencies above 7500 Hz. The spectrogram of a unit-selection attack, figure 26, shows regions where all frequencies are occupied. Based on



(a) Human speech signal

(b) Transition in human speech signal

Figure 23: Example of a human speech signal and transitions



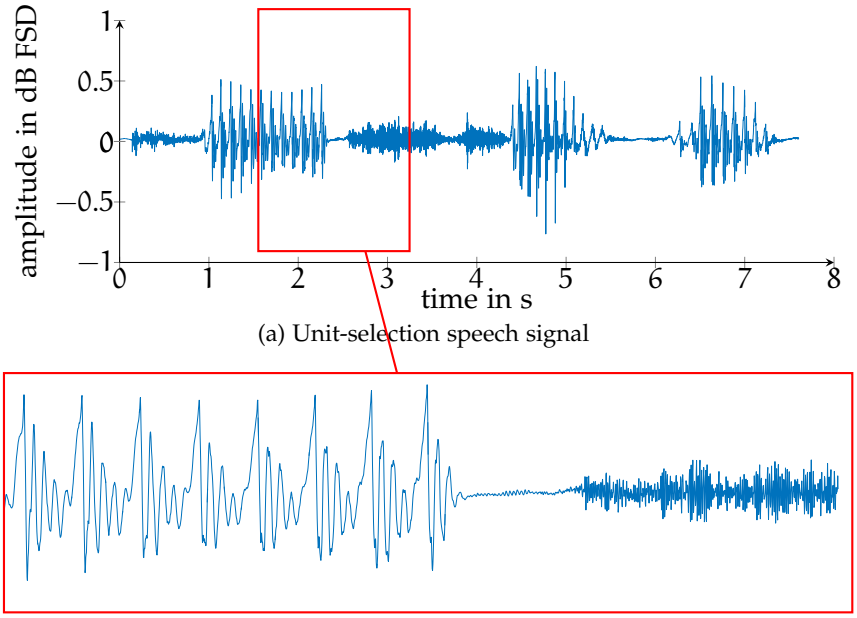
(a) Spectrogram of human speech signal

(b) Spectrogram of transition in human speech signal

Figure 24: Spectrogram of a human speech signal and transitions

these observations, multiple possibilities in detecting a discontinuity in audio signals arise.

As a non-continuous signal causes higher frequencies in the frequency domain, a Fourier transformation of the whole signal should provide bigger amplitudes in higher frequencies as a smooth human signal does. The frequency spectrum of a unit-selection attack should be distinguishable from the frequency spectrum of human speech.



(a) Unit-selection speech signal

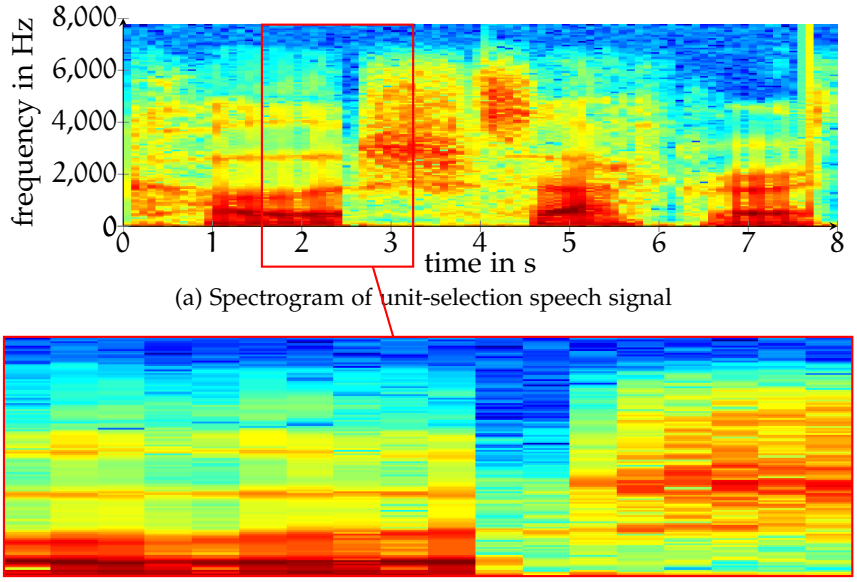
(b) Transition in unit-selection speech signal

Figure 25: Example of a unit-selection speech signal and transitions

The Fourier transformation results in a two dimensional vector over frequency and amplitude. In purpose of creating a score, the sum over the derivation is calculated.

4.1.2 *Spectrogram-based Detection*

Another countermeasure is motivated by the assessment of the spectrogram of the audio files. As figure 26b points out, the spectrogram of unit-selection attacks contains non-natural bars through all frequencies. For the purpose of calculating the spectrogram, multiple parameters have to be defined. The signal has to be divided into windows of fixed length, a defined overlap of the windows is common. In order to reduce the leakage effect of the Fourier transformation, a window function is applied to the signal [38]. Hamming or Gaussian windows are common for speaker recognition algorithms. In general, an algorithm for the transformation into the frequency domain has to



(a) Spectrogram of unit-selection speech signal

(b) Spectrogram of transition in unit-selection speech signal

Figure 26: Spectrogram of a unit-selection speech signal and transitions

be selected. A fast method is the STFT with FFT [42, p. 294]. For the STFT method, a granularity of the determination of frequencies has to be defined. Smaller windows increase the accuracy of the result, but also the expense of the transformation and the size of the output.

State-of-the-art speaker recognition systems utilize spectrograms for the generation of MFCCs [7]. Common parameters for the generation of the spectrogram for MFCCs are a window size of 20 ms, a step size of 10 ms, a Hamming window [79] and a resolution of 256 frequency bins. The window size is chosen in order to be able to account for changes in the human voice.

As this approach aims at the detection of bars through all frequencies, much smaller window sizes around 5 ms can be chosen. Accounting for the smaller windows, an overlap of 3 ms is chosen, persisting the window to overlap ratio. This results in a corresponding step size of 2 ms. For the window, a Gaussian or Hamming function should be considered, as it fortifies the appearance of the bars.

There are at least two possible ways of detecting the bars in the spectrogram. First is to monitor all frequencies higher than a threshold, for instance 15 600 Hz for a 16 000 Hz signal. As the spectrogram of human speech does not occupy these regions, the maximum of the sum of the energy of these frequencies can detect an unit-selection attack. There are further possibilities of fusing the energies, for example by the product, the square sum or a weighted sum.

As the sudden change of the signal in unit-selection attacks affects the whole frequency band, a more sophisticated approach may take all frequencies into account. The product over all energies per window would be zero for windows with unoccupied frequencies, whereas windows with all frequencies occupied should result in a huge product. In order to raise the influence of the higher frequencies, a linear or exponential weighting can be applied to the spectrogram. In the case of samples recorded with 16 000 Hz, a spectrogram with 8000 frequencies can be generated. A multiplication of 8000 values can lead to extremely high values which are hard to calculate and store. Hence a previous binning of the frequencies is mandatory. For the purpose of initial analysis, a number of 160 bins seems to be suitable.

The STFT results in a three dimensional vector over time, frequency and amplitude. In a first step, the amplitudes are merged over the frequencies. As the focus is put on higher frequencies, lower frequencies are omitted. Possibilities for merging are for example the sum, RMS or product. As the change of high frequencies is of interest, the derivation of the vector is calculated. In order to obtain a scalar, the sum over the vector is conducted.

4.1.3 *Wavelet-based Detection*

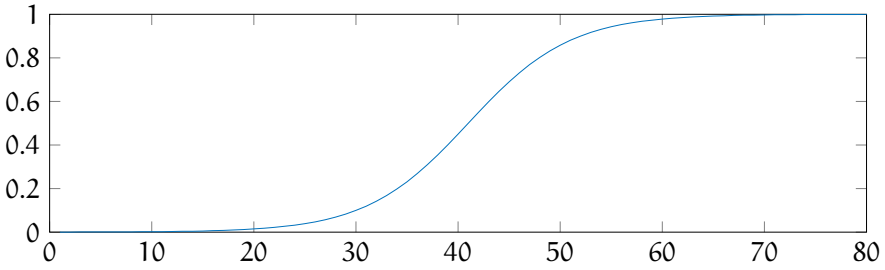
Another possibility of frequency analysis is the wavelet transformation. As shown in section 2.4.3, the main difference between STFT and wavelet transformation is the relation between time and frequency resolution. The wavelet transformation provides a higher time resolution for higher frequencies. As the detection of unit-selection attacks focuses on the upper frequency band, this transformation seems to be promising for detecting the unit-selection specific changes.

In practice, the DWT is used as an implementation for the wavelet transformation. As shown in section 2.4.3, the DWT can be understood as a bandpass filter. Each iteration provides a new level of detail. The number of reasonable detail levels will be examined in this thesis, also a fusion of detail levels can be used to improve the detection algorithm.

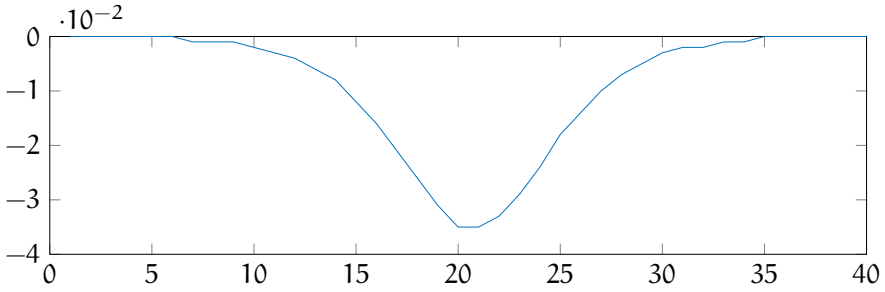
The result of the DWT is a two dimensional vector over time and amplitude. As only the detail levels of higher frequencies are considered, the vector contains the high-pass filtered acoustic signal. In order to detect changes in the high frequency band, the signal is derived, with the assumption, that unit-selection samples contain more changes in higher frequencies, thus the length-normalized sum over the signal is calculated. The score of unit-selection attacks is expected to be higher, than the score for bona fide samples.

4.1.4 *Edge-Detection-based Detection*

The wavelet transformation, in particular the DWT is for instance known for edge detection in the field of computer vision [80, p. 15]. A possibility is be to bend an edge detection algorithm to detect the transitions in unit-selection samples. The wavelet transformation is



(a) Sigmoid function



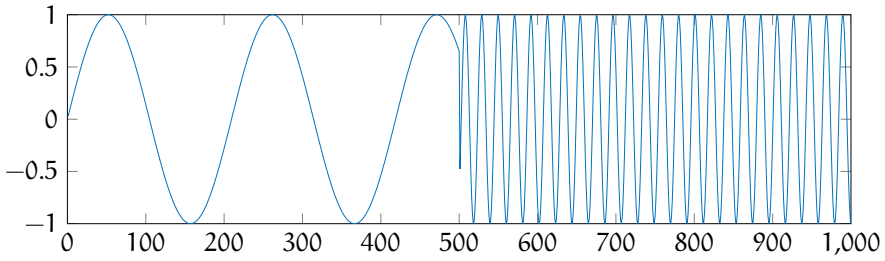
(b) First detail level

Figure 27: Wavelet Transformed of sigmoid function

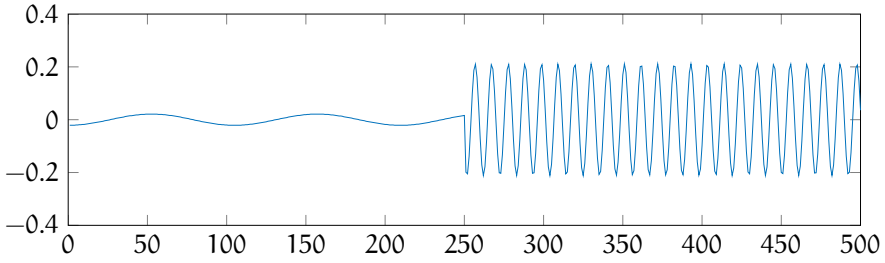
utilized as an edge detector, in order to detect fast falling or ascending sections of the signal representing the image. In the details of a wavelet transformation, the fast changing parts of the signal result in higher amplitudes [81]. Figure 27 exemplary demonstrates an edge detection. The sigmoid function symbolises an edge, for example the grey scale changes from dark to light. The first detail level of a wavelet transformed, figure 27b, shows a peak, where the sigmoid function changes the most.

Transitions in concatenated unit-selection samples are not directly comparable to edges in images. They cannot be detected by abrupt changes in the signal itself, but by changes in the frequency of the signal. Figure 28 depicts an abrupt change of a oscillating signal. The first detail level of the wavelet function works as a high pass filter to the original signal. The transition does not generate a detectable peak in the wavelet transformed signal.

Continuative approaches may apply edge detection on the spectrogram. As illustrated in figure 26a, the transitions in unit-selection samples cause edges in the spectrogram. A wavelet based approach is



(a) Transition in signal



(b) First detail level

Figure 28: Wavelet transformed of transition

expected to be capable of detecting these edges, and thus be utilized for generating features for unit-selection attack countermeasures. Due to the further research required by this approach and the tight timing constraints of the thesis, this feature will not be processed further in this thesis.

4.2 EXPERIMENTAL SET-UP

In general, for the development of PAD systems, a database with according attacks is needed. For the task of detecting unit-selection attacks, only the S10-attacks provided by the evaluation set of the ASVspoof are available. As the evaluation set should remain unseen during the development process, additional unit-selection attacks are necessary in order to create distinct data sets.

4.2.1 *Databases for Unit-Selection Voices*

In order to generate an independent database, a certain amount of unit-selection samples has to be produced by a set of unit-selection voices. Every unit-selection voice is created by using a set of bona fide samples of one subject. As the database will be used for developing a method for distinguishing bona fide and unit-selection samples, a sufficient number of subjects should not be used for the creation of unit-selection voices. In order to find a proper development database for the purpose of examining unit-selection countermeasures, speech synthesis and speaker recognition databases are discussed regarding

their suitability for the generation of unit-selection voices, in particular ARCTIC, MOBIO, YOHO and Open Speech Data Corpus for German.

ARCTIC[1] data from CMU contains seven hours of speech, but only one subject. The unit-selection voice generated with this database is able to produce comprehensible sentences. For the database the ARCTIC data can not be used, as it only contains one subject.

Another database deserving attention is the MOBIO database[2]. MOBIO is a bi-modal database covering video and audio. 152 subjects from 6 different sites have been captured in 6 sessions. The sessions consist of short response questions, free speech questions and the reading of a predefined text. The database has no explicit transcription files, but the answers to the short response questions and the predefined text are well known. After an adjustment of the data, a total duration of less than 45 minutes per subject remains. Most samples of one subject contain the same sentences, for example the answer to "'What is your name?"' does not vary. Thus the generated unit-selection samples are prone to be of poor quality, since the variety of units is reduced.

The next database to be examined is the YOHO Speaker Verification corpus[3]. It contains 138 subjects with 136 utterances in 14 sessions. The utterances consist of two-digit numbers spoken continuously in sets of three. The samples are named with the numbers read, so a transcription of the samples can be derived. Due to the high number of samples, the database reaches a total duration of approximately 90 minutes of speech per subject. As the samples solely consist of numbers, the amount of diphones is lower than for common text.

A differing database is the Open Speech Data Corpus for German from TU-Darmstadt[4]. MOBIO and YOHO are designed for speaker recognition tasks, whereas this database is designed for speech synthesis. The data has been recorded in a clean environment with a constant distance between subject and microphone. In order to reduce speaking errors and artefacts, the samples are reworked. The database contains utterances of 180 subjects, reading German Wikipedia, protocols from European Parliament and some individual commands. This leads to a total of around 4 hours of speech per subject. The database provides a full transcription of the samples.

1 http://www.speech.cs.cmu.edu/cmu_arctic/packed/cmu_us_slt_arctic-0.95-release.tar.bz2
2 https://www.idiap.ch/dataset/mobio
3 https://catalog.ldc.upenn.edu/LDC94S16
4 https://www.lt.informatik.tu-darmstadt.de/de/data/open-acoustic-models/

Table 2: Databases for unit selection voices

Database	Subjects	Speech Per Subject	Transcription	Language
ARCTIC	1	ca. 7 h	Yes	English
MOBIO	150	<45 min	partial	English
YOHO	138	<90 min	derived	English
G Speech Data	179	<4 h	Yes	German

Table 2 benchmarks the discussed databases in terms of suitability for unit-selection attacks. As the German Speech Data Corpus promises the most success for generating high quality unit-selection voices, it has been chosen for the creation of the development set. In order to achieve stable results, a development set with a balanced ration of bona fide and attack samples is eligible.

The creation of a unit-selection voice is time-consuming and laborious, whereas the generation of unit-selection samples with the voice is very fast and only limited by the number of sentences available. As the quality of the unit-selection voice depends on the time of speech material used for training, the 20 subjects with the highest overall sample duration are utilized to generate 20 unit-selection voices.

4.2.2 *Sentences for Unit-Selection Attacks*

After the creation of unit-selection voices, 12 950 bona fide speech samples of 169 subjects remain unaffected. In order to achieve a balanced development set, ca. 15 000 unit-selection samples have to be generated. As 10 unit-selection voices are generated, each voice may produce 1 500 samples. To avoid unintended dependencies, each voice is assigned with its own sentences.

The source for the german sentences is a Dump of the Wikipedia-Database, accessible via Wikimedia[5]. The archive contains all german Wikipedia-Pages of the 26.12.2015. The Wikipedia Extractor of the Università di Pisa[6] is utilized in this thesis in order to obtain plain text transcription.

The resulting text files were separated into sentences. Afterwards the sentences are sanitized. Special characters and short sentences are removed. From the remaining sentences, 15 000 are randomly selected and distributed in equal parts among the unit-selection voices.

5 https://dumps.wikimedia.org/dewiki/20151226/dewiki-20151226-pages-articles.xml.bz2

6 http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

4.2.3 *Protocol for the Unit-Selection Database*

The creation of a unit-selection speaker takes at least two hours calculation time per subject. Due to timing constraints, a set of 20 unit-selection speakers was created. In this thesis, a protocol for a unit-selection database is derived, providing a distinct development and evaluation set. For a second evaluation on an independent dataset, the ASVspoof database is utilized.

4.2.3.1 *Development Set*

The development set is used for developing unit-selection detecting algorithms. The subset of human speaker and unit-selection attack does not need to contain the same subjects.

In order to be able to find a way of discriminating bona fide and attack samples, the aim of the development set is a large number of bona fide and attack samples. The corpus features 179 subjects. 20 subjects were used for the generation of unit-selection attacks. 159 subject are remaining for the bona fide subset. A total of over 14 000 bona fide samples is achieved. 10 of the 20 unit-selection speakers were used to create the development set. In order to achieve comparable sized bona fide and attack subsets, 15 000 attack samples were created.

4.2.3.2 *Evaluation Set*

The evaluation set is used to determine the performance of the developed algorithms. Not only the capability of the algorithm of discriminating bona fide and attack samples, but also the baseline performance of a SIV-system and the performance of the same system against unit-selection attacks is of interest.

Therefore, the dataset has to consist of an enrolment- and a verification-subset with bona fide samples and an attack subset containing unit-selection attacks. 10 unit-selection voices are trained for this subset. 50 bona fide samples of each subject are not used for the training and divided into 40 enrolment and 10 verify samples. Eachunit-selection voice generates 10 attack samples. Summarized, the evaluation dataset consists of ten subjects with each 40 reference, 10 probe and 10 attack samples. The resulting database is displayed in table 3.

4.2.3.3 *Second Evaluation Set*

In order to determine the stability of the developed algorithms, the unit-selection attacks of the ASVspoof database are utilized. ASVspoof provides 9 404 bona fide samples and 18 398 unit-selection attacks.

Table 3: Protocol for unit-selection database

Subset	Enrol	Verify	Attack
Development Set	14 088 bona fides		14 945
Evaluation Set	400	100	100
ASVspoof S10	9 404 bona fides		18 398

4.2.4 *Algorithms*

Experiments are carried out investigating the performance of the three methods for frequency-based unit-selection attack detection proposed in 4.1. In order to reach comparable results, the signals are adapted. First, the silence has to be removed. This can be done by VAD. In this approach a basic energy-based silence-detection [35] is applied. With the silent parts removed, the amplitude of the signal has to be normalized in order to reach comparable spectrograms, in this approach a maximum-normalization was employed.

Fourier-Based detection: For the Fourier transformation of the signal the FFT implementation of MATLAB was utilized. In order to reduce the leakage effect of the transformation a hamming-window was applied to the signal. The score S is calculated according equation 39:

$$S = \frac{1}{N} \sum_{k=1}^{N} F'_d(k). \tag{39}$$

Whereas

$$F_d(k) = \sum_{n=0}^{N-1} f(nT_a) \cdot e^{\frac{-j2\pi kn}{N}} \tag{40}$$

and $N = 8\,000$, as a signal with 16 000 Hz is analysed.

Spectrogram-Based detection: The spectrogram based approach utilizes the spectrogram implementation of MATLAB, which employs the STFT. The window size for the STFT is set to 80, which corresponds 5 ms. In order to reduce the leakage effect of the Fourier transformations, a hamming window is applied to each frame. Due to the fragmentation of the signal into frames and the usage of a window function, transitions between two frames may remain undetected. To avoid this problem, a window overlap of 75% is utilized. The resolution of the STFT is set to 10% of the signal frequency.

The chosen configuration for the STFT results in a three dimensional vector over time, frequency and amplitude. In order to obtain a score, the dimensions have to be reduced. The score S is calculated as

$$S = \sum_{m=0}^{M} \left(\sum_{k=K}^{N} F^{\gamma}(m,k) \right)', \tag{41}$$

whereas M has to be the number of frames of the STFT. As changes in the higher frequencies are of interest, the lower frequencies are discarded and the higher frequencies are derived. The limit for the frequencies, K, is set to 100, 500 and 1000 which corresponds to the band from 15 800, 15 000 and 14 000 to 16 000 Hz. The STFT, $F^{\gamma}(m,k)$, is defined in equation 23. The sum over the frequencies can be replaced with the RMS or the product. As the utilization of the product easily results in infinite or zero values, only sum and RMS were tested.

Wavelet-Based detection: For the wavelet-based approach, the DWT implementation of MATLAB is utilized. Three different wavelets are tested: Haar, Daubechies 1 and Daubechies 10 [46]. In a first attempt the details of the first three iteration levels are analysed, later the examination is extended to the sixth iteration.

As described in section 2.4.3, the DWT can be considered as bandpass filter. Therefore, the outcome of the DWT is a two dimensional vector over time and amplitude. As the changes in the high frequencies are of interest, the derivation of the details are considered. In order to obtain a score, the weighted sum of the derived details is calculated.

4.3 EVALUATION OF BASIC APPROACHES

The performance of the proposed algorithms is examined on the development set defined in section 4.2.3.1. A PDF is used to visualize the score distribution for bona fide and attack samples. In order to create a comparable result of the classification performance of the algorithms, the EER and DET diagram of APCER and BPCER are utilized.

The proposed basic approaches assume more occurrences of frequency changes on unit-selection signals than on human speech, see section 4.1. As the score represents a measure for presentation attacks, it will be referred to as PA score.

4.3.1 *Results for Fourier-based Detection*

The PA score generated by the Fourier transformation approach shows a difference for bona fide and attack samples. In the PDF, figure 29, a different distribution of the scores is visible.

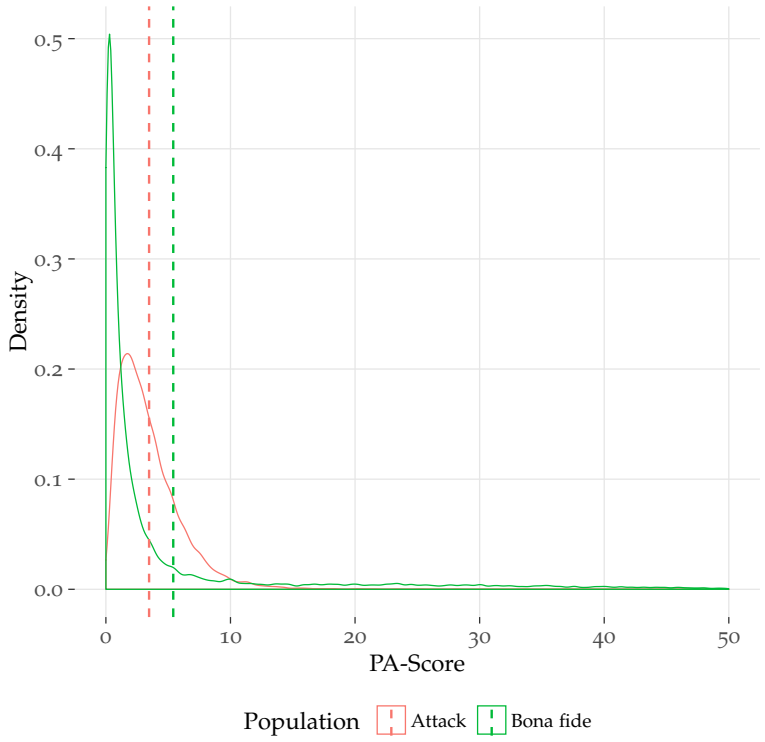


Figure 29: PDF for basic Fourier transformation based approach

The mean of both distributions differs, but a large overlap of both distributions degrades the performance of the tested PAD algorithm. The DET in figure 30 visualizes the performance of the algorithm,

yielding an EER of 33.4%, which is comparable to the unit-selection detection performance of many ASVspoof submissions but not applicable.

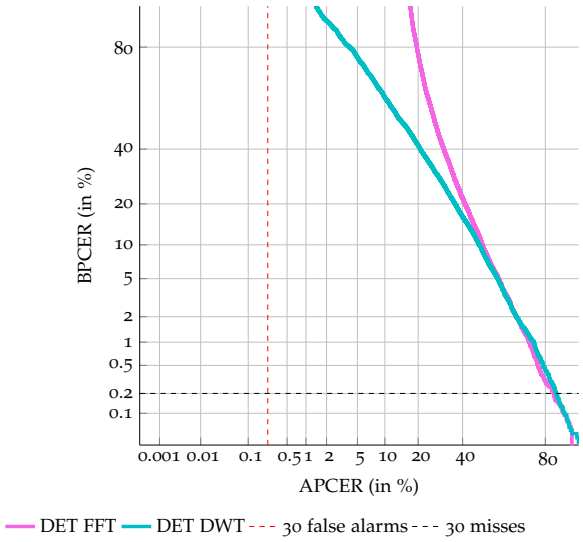


Figure 30: DET for basic Fourier transformation based approach and wavelet based approach with Haar wavelet and 5 iterations

4.3.2 *Results for Spectrogram-based Detection*

The proposed basic approach of employing a scalar representation for spectrogram results in poor performance which can be attributed to the variability of calibrateable spectrogram parameters. The PDFs, figure 31, of bona fide and attack samples are indistinguishable.

Table 4: EER for PAD with spectrogram based approach

Frequencies	**100**	**500**	**1000**
RMS	43.6	41.6	42.5
Sum	43.8	41.6	41.9

The EER for the different configurations are displayed in table 4. The best performance achieved is an EER of 41.6% for 500 frequencies for RMS or Sum. The calculation of the spectrogram feature has a lot of variables to tune. Window size, overlap and frequency resolution are parameters that can be changed at STFT level. The fusion of frequencies can be done by more advanced methods like weighted sums and the range of used frequencies can be deeper explored. Due to the bad initial performance, the approach of spectrogram based analysis for PAD will not be traced further in this thesis.

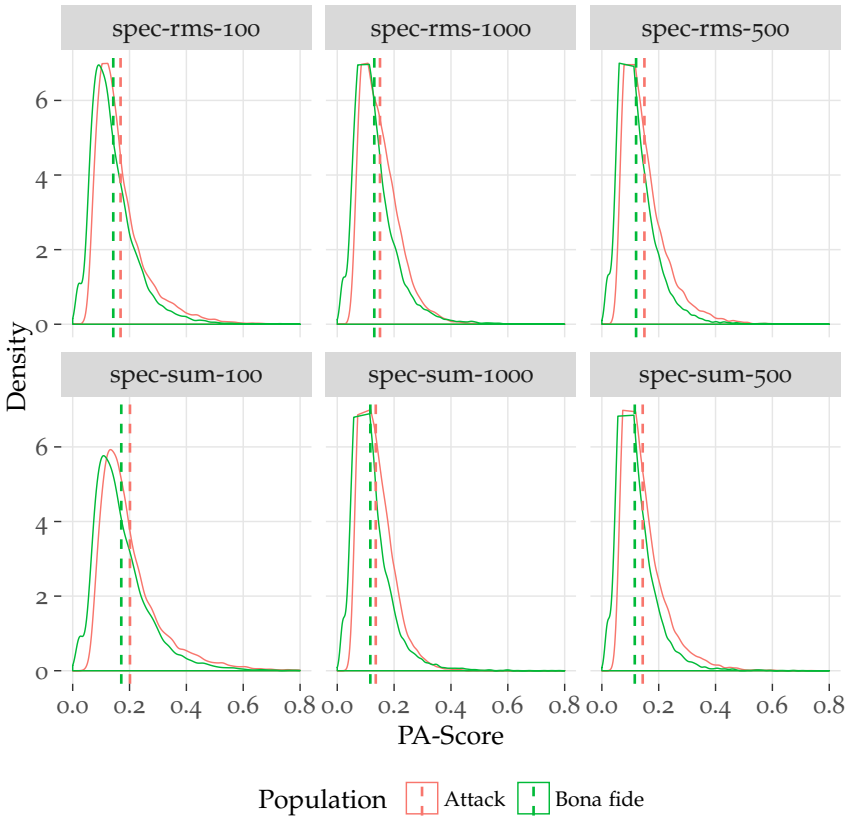


Figure 31: PDF for basic spectrogram based approach

4.3.3 *Results for Wavelet-based Detection*

The wavelet-based detection seems to be the most promising approach. For the transformation, three wavelets are tested: two Daubechies wavelets (DB1 and DB10) and a Haar wavelet. The resulting vector is summarized to a scalar. The PDFs of PA scores of bona fide and attack samples of the first three iterations is displayed in figure 32.

The difference between the BD1 and Haar wavelet is negligible. The highest discrimination between attack and bona fide samples can be made for DB1 and Haar wavelets in third iteration.

The resulting EERs for different wavelets and iterations are listed in table 5, the lowest EER achieved is 33.2%. Contrary to the assumption, later iterations, displaying lower frequency bands, yield a higher distance of the PDFs. As the EER improves for deeper detail levels, further iterations are analysed in order to find the optimum. The wavelets achieve a minimum EER at the fifth iteration. At sixth iteration, the performance is declining. Figure 30 shows the DET function for the configuration causing the lowest EER on the development set.

Table 5: EER for PAD with wavelet based approach

Iteration	DB1	DB10	Haar
1	40.2%	44.4%	40.2%
2	33.4%	33.4%	33.4%
3	33.2%	39.1%	33.2%
4	31.4%	36.2%	31.4%
5	29.8%	30.6%	29.7%
6	44.5%	38.3%	44.6%
Fusion 3-5	33.1%	37.4%	33.1%
Fusion 1-5	40.8%	40.8%	42.0%

As the bandpass filter changes for each iteration, the returned signal for each iteration contains different informations. Therefore, the merge of the results of multiple iterations can achieve an improve in performance of the PAD algorithm. Two approaches of fusion are made. The fusion is applied on the details of the iterations, as method sum is used. The three most promising iterations, three, four and five, are fused and in another attempt the first to fifth iteration.

No further gains are yielded by this fusion compared to single detail level approaches. Table 5 shows the EERs for both fusion approaches with the different wavelets. The fusion is approximately as bad as the worst iteration used for the fusion.

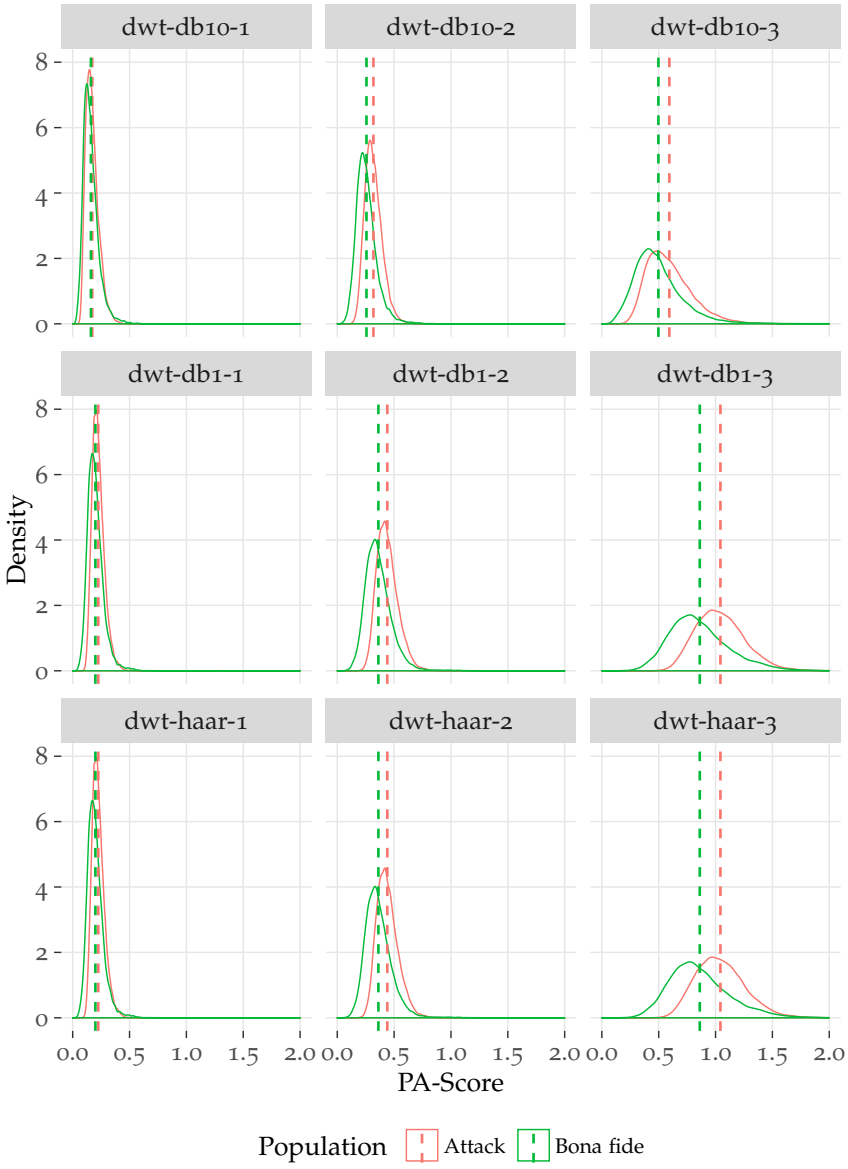


Figure 32: PDF for basic wavelet based approach

4.4 IMPROVEMENTS OF FREQUENCY-BASED DETECTION

The evaluation of the basic PAD algorithms shows the need for more advanced countermeasures. The best result achieves an EER of 29.7%, which is not satisfactory for practical scenarios. Due to the simple reduction of the feature vector to a score by building the sum, many informations are lost. In order to explore Fourier, spectrogram, and wavelet-features more in depth, non-trivial comparison approaches are investigated, in particular SVMs and GMMs. Three different frequency based features are examined in combination with the machine learning techniques.

4.4.1 *Design of Feature Vectors*

For the purpose of SVM and GMM-based comparisons, coherent feature spaces need to be designed, which allow class modelling. A simple method for generating fix-length vectors would be to define a specific length for the samples. Longer samples are truncated, shorter samples are padded to the desired length. A big disadvantage of this method is, that information is lost by truncation and fragments are added by padding. In the field of speaker recognition and in the state-of-the-art of unit-selection detection, the employment of the fix-dimensional MFCC-feature vectors is common, where a decorrelation and dimension reduction is performed by employing a DCT. By transforming the signal into the frequency domain, a fix-dimensional feature space can be established. The frequency domain is time-independent, each outcome of the Fourier transformation has the same length. In the thesis the Fourier transformation is employed instead of the DCT.

In general, long feature vectors may comprise more information. But a machine learning algorithms like GMM or SVM needs more samples to train with larger feature vectors. The number of training samples is limited, so a trade-off for the feature length has to be found.

Fourier-Based Feature Vectors: The basic approach with Fourier transformation is already time independent as the FFT is employed. By changing the resolution of the FFT the resulting feature space dimensionality is varied. A lower resolution contains less information, but produces shorter feature vectors.

Wavelet-Based Feature Vectors: The outcome of the wavelet transformation is linear dependant from the length of the audio sample. As shown in section 2.4.3, the Fourier transformation provides a lossless transformation of any signal into the frequency domain. If a FFT is applied to the outcome of the wavelet transformation, it produces

a representation of the wavelet transformation as an equal length vector. The length of the feature vector can be varied by the resolution of the FFT.

Due to the Fourier transformation, both feature vectors contain complex values. In order to avoid complications with the machine learning algorithms, the complex magnitude, $|a + bi|$ of the values is calculated as

$$|a + bi| = \sqrt{a^2 + b^2}. \tag{42}$$

4.4.2 *Subsets for Machine Learning*

The machine learning algorithms examined in this thesis are SVM and GMM. SVM was chosen, as it represents a well established machine learning algorithm which provides binary classification, GMM is a common classification method in speaker recognition [13, 24, 28].

In general, machine learning algorithms are based on the training on data samples. As a testing of the trained SVM and GMM is mandatory, the development set defined in section 4.2.3.1 has to be subdivided. In order to avoid data snooping, the subjects in training and test set should be distinct. As shown in table 3, the development set consists of 12 950 bona fide and 14 945 attack samples. The attacks are generated from 10 subjects, the bona fide samples are from 159 subjects. 70% of each population of the development set are selected for the train set, the remaining 30% are used for testing.

Table 6: Test and training set for machine learning

Subset	Bona Fide Samples	Attack Samples	Attack Subjects	Bona Fide Subjects
Train	10 343	10 461	7	111
Test	3 745	4 484	3	48
Development Set	14 088	14 945	10	159

4.4.3 *Training the Machine Learning Algorithm*

According to section 4.3, the Fourier and wavelet based approaches are the most promising. Therefore these two are further examined with machine learning algorithms. The Fourier based approach is tested with frequency resolutions from 100 to 3 000. The wavelet based approach is tested for the DB1 wavelet with five iterations, as well as for the best fusion approach with iteration three to five.

Two machine learning approaches are examined in the thesis. For a simple binary classification a SVM is utilized. As SVMs are known for good pattern recognition performance [20, p. 1504], it is likely that they are capable of determining characteristic patterns for distinguishing bona fide and attack samples. Following the assumption, that Fourier based feature spaces comprise linear segregable populations, linear SVM kernels may yield adequate performances. In addition to the SVM approach, a GMM was trained. GMMs are known for good performance in speaker verification scenarios. Due to the limited amount of training data, a GMM with 16 components is chosen.

4.5 EVALUATION OF MACHINE LEARNING APPROACHES

Due to the subdivision of the development set, the set of samples used for the evaluation of the trained SVMs and GMMs is 30% of the set used for evaluating the basic approach. As the test-set represents a subset of the development set, the results should be nearly comparable. In order to achieve comparable scores, EER and DET will be utilized to exemplify the PAD performance of the improved algorithms.

4.5.1 *Results for PAD with Machine Learning*

The different EERs of the algorithms observed at different frequency resolutions are depicted in figure 33. The best EER of 5.0% can be achieved with a feature created by the wavelet approach with fusion of the third, fourth and fifth iteration and FFT with 600 frequencies.

The performance of the GMMs is in general not as good as the performance of the SVMs. Further tuning of the parameters of the GMMs may improve the performance. The performance for all algorithms degrades, if the number of frequency bands exceeds 1000. It can be assumed, that the feature space dimensions reach a limit, where the machine learning algorithms are not capable of distinguishing bona fide and attack samples, with the number of samples available for training. The EER of the approaches utilizing the wavelet algorithm are increasing faster than the algorithms utilizing the basic FFT. Table 7 summarizes the best-performing observed configurations for each algorithm.

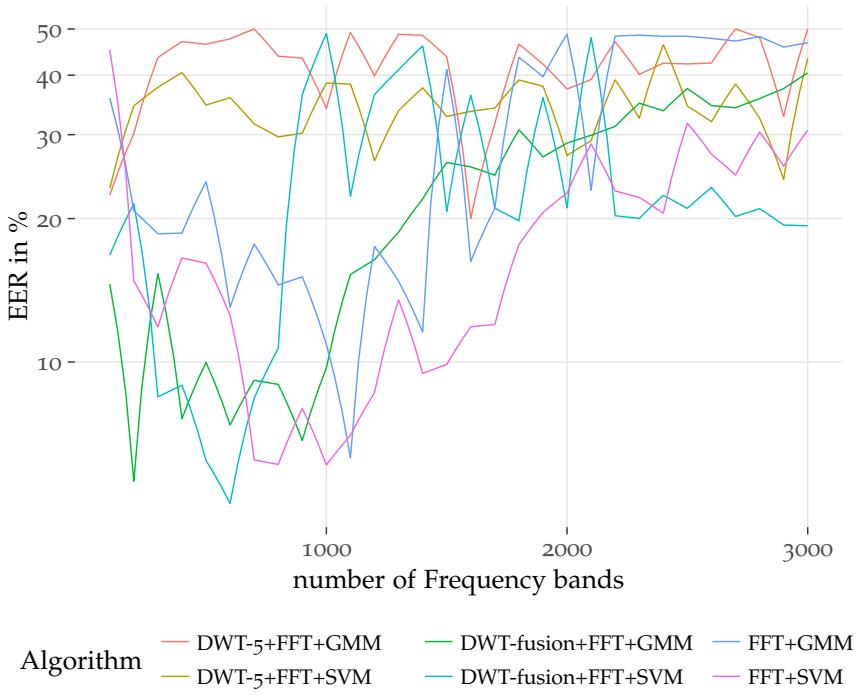


Figure 33: EER for machine learning approach with different frequency resolutions

The DET plots of the configurations listed in table 7 are displayed in figure 34. There are no fundamental differences in the shape of the DET plots of the FFT based approach and the DWT based approach with fusion. The performance of DWT without fusion is far beyond the others. All plots are vertical straightened, which influences the behaviour of the PAD algorithm if the threshold is moved from the EER. A lower BPCER can easily be achieved. With the DWT approach with fusion and SVM, for example, an BPCER of 1% results in an APCER of 10%. A lower APCER however results in a fast increase of the BPCER. An APCER of 1% would lead to a BPCER of nearly 80%.

Table 7: Configuration for best EER

Algorithm	**EER**	**Number of Frequencies**
DWT-fusion+FFT+SVM	5.0%	600
DWT-fusion+FFT+GMM	5.6%	200
FFT+SVM	6.1%	1000
FFT+GMM	6.3%	1100
DWT-5+FFT+SVM	23.1%	100
DWT-5+FFT+GMM	20.0%	1600

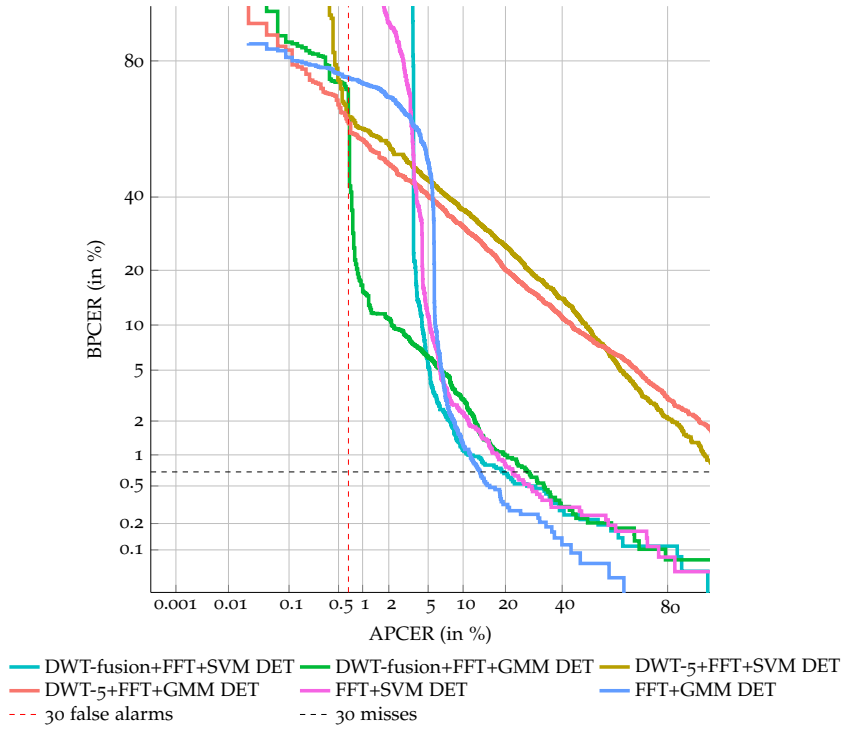


Figure 34: DET plots for configurations with best EER on development set

4.5.2 *Results for Evaluation Set and ASVspoof*

In general, machine learning algorithms carry the risk of overfitting. In this case, the results on the training set are much better than on other sets. The best configurations, listed in table 7, are tested against the two evaluation sets, defined in table 3. An almost constant performance of the algorithms would support the hypothesis of universality. As the data available for evaluation in the development set and both evaluation sets differ, a comparison of the calculated EERs and DET functions is not exact. Additional to the difference in the sample number, the ratio between bona fides and attack samples depends on the utilized set.

As shown in table 3, the evaluation set has a small number of samples. According to the Rule of 30 [82], to be 90% confident that the true error rate is in a confidence interval of ±30% of the measured error rate, at least 30 errors have to be detected. Lines for 30 bona fide and attack presentation classification errors are displayed in the DET plots. Whereas the DET plots for development set and ASVspoof data stay far beyond the 30 errors lines, the DET plot for the evaluation set falls below the *Rule of 30*. This has to be considered when examining the results.

Table 8: Best configurations evaluated with evaluation set and ASVspoof

Algorithm	EER Eval-set	EER ASVspoof
DWT-fusion+FFT+SVM	7.1%	11.7%
DWT-fusion+FFT+GMM	15.0%	24.6%
FFT+SVM	8.5%	22.6%
FFT+GMM	9.5%	27.7%
DWT-5+FFT+SVM	27.0%	11.7%
DwT-5+FFT+GMM	40.1%	45.7%

Table 8 benchmarks the EERs for the best algorithms of the previous approach. SVM and GMM are the ones trained on the development set data. A repeated training on the evaluation sets may reinforce the performance, but as in a real-world scenario the attacks are not known, a repeated training would distort the results.

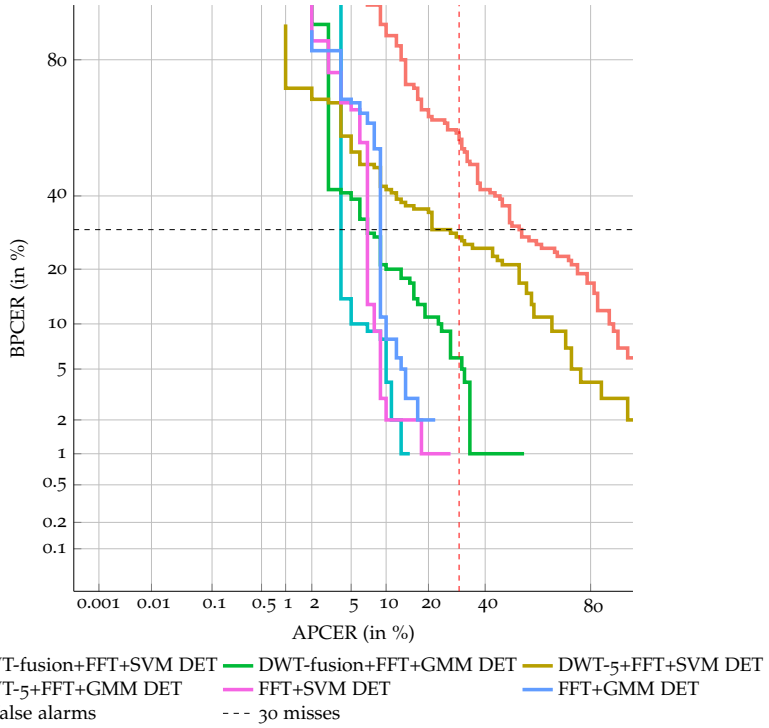


Figure 35: DET plots for configurations with best EER on evaluation set

Independent from the algorithms, the performance of the SVM is less affected than the performance of GMM. The best algorithm of table 7 remains the best for the evaluation sets. The performance of the DWT approach without fusion is only slightly affected by different data sets. The EER for the development set and ASVspoof remained the same. A possible explanation for this behaviour is the small number of 100 frequencies analysed in this configuration.

Figure 35 visualizes the DET performance of the algorithms on the evaluation set. The function is rather steppy than the one shown for the development set. This is caused by the much lower number of samples available in the data set. The shape of the DET function is nearly the same, compared to the development set.

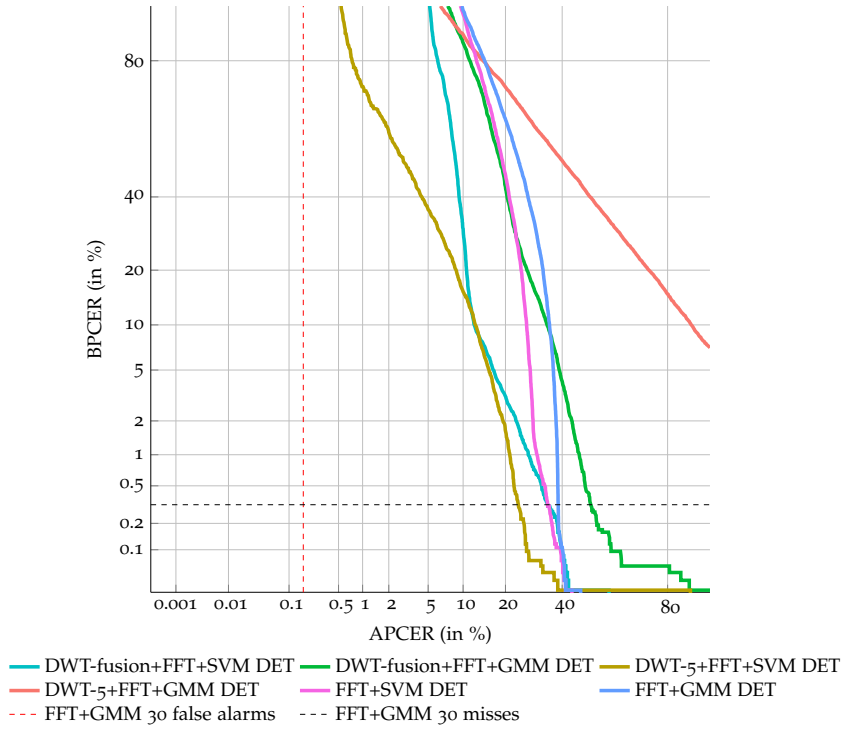


Figure 36: DET plots for configurations with best EER on ASVspoof S10 attacks

Figure 36 visualizes the DET performance of the evaluation with the ASVspoof S10 attack. The EER of the DWT algorithm with fusion and SVM increases by 6.7 percent points, which doubles the error rate. All other algorithms, but the DWT with five iterations and SVM, rise over 20%. The performance of the DWT with five iterations and SVM surprises, as it doubles compared to the evaluation set. The DET function of the algorithm is the best in this data set, the EER is the same as the DWT approach with fusion.

4.5.3 *Results with Larger Training Sets*

In general, the performance of machine learning algorithms strongly depends on the amount of data used for training. An approach of increasing the performance of the introduced algorithms was made by increasing the training data. For this purpose, the complete development set is utilized for the training of the algorithms. The new

trained classifiers cannot be evaluated with the development set, but an evaluation with the evaluation set and ASVspoof data is still possible.

Table 9: Best configurations evaluated with evaluation set and ASVspoof, algorithm trained on full development-set

Algorithm	**EER Eval-set**	**EER ASVspoof**
DWT-fusion+FFT+SVM	11.3%	13.8%
DWT-fusion+FFT+GMM	14.7%	20.9%
FFT+SVM	13.0%	24.9%
FFT+GMM	12.0%	23.2%
DWT-5+FFT+SVM	27.0%	11.1%
DwT-5+FFT+GMM	41.9%	46.9%

The EERs of the new trained algorithms are listed in table 9. In general, the performance of SVM based algorithms decrease, except the DWT approach without fusion. For the GMM algorithms, the DWT-fusion approach, improves the performance, in particular on ASVspoof S10 data.

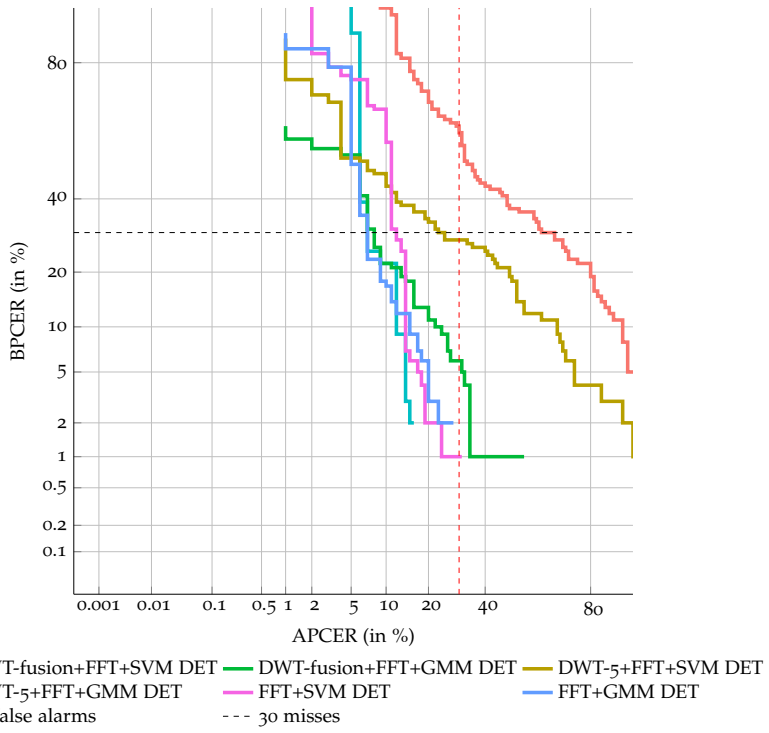


Figure 37: DET plots for configurations with best EER on evaluation set, Algorithm trained on full development-set

Figure 37 visualizes the DET functions of the new trained algorithms. In general, the new training did not effect the shape of the functions. It is difficult to determine a difference between the DET functions displayed in figure 35 and the new DET functions in figure 37.

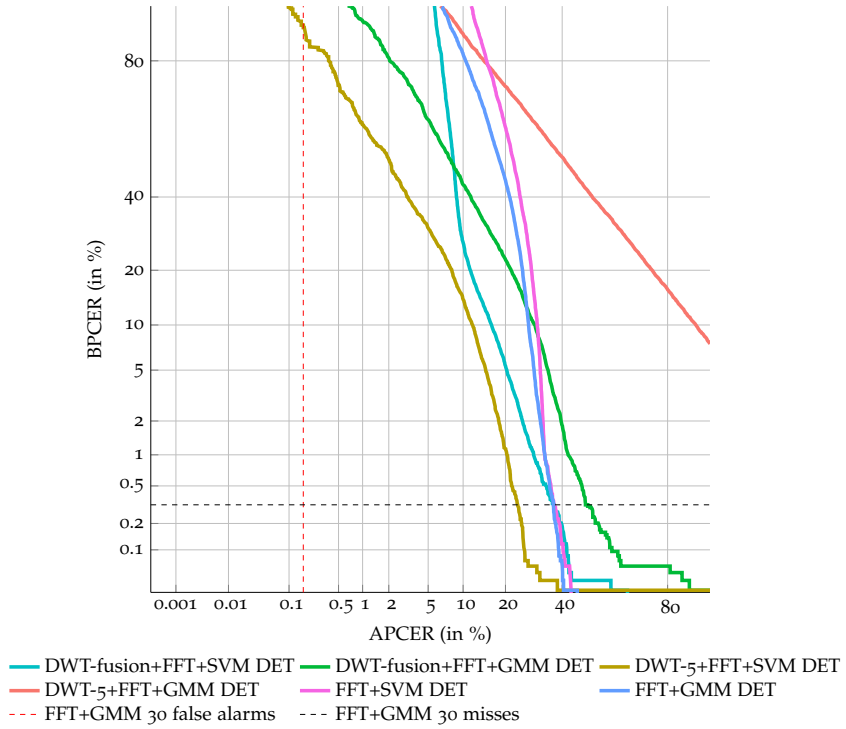


Figure 38: DET plots for configurations with best EER on ASVspoof, Algorithm trained on full development-set

Figure 38 shows the DET functions for the evaluation of the new trained algorithms with the ASVspoof attacks. Due to the broader training set, the DET for the DWT approach of the fifth iteration and SVM outperforms all other systems amongst all operating points. Especially for high BPCER the APCER is lowered.

4.6 SUMMARY

The examined SIV baseline system is not capable of detecting an unit-selection attack. A preceding PAD is required. Conventional PAD algortihms employ features conventionally utilized in speech and speaker recognition. In this thesis, it is assumed, that information is lost by extracting common features, due to the employed filter banks. Thus, three different frequency analysing methods for unfiltered signals are proposed and examined based on Fourier spectrum, spectrogram, and wavelet transformation. In order to achieve a detection score, the sum of the derivation of the tested algorithms is calculated. However, non satisfiable results are yielded, indicating a non-trivial

problem. The spectrogram based approach is not capable of distinguishing bona fide and attack samples. The Fourier spectrum and wavelet transformation approaches show two differing distributions for the samples with a huge overlap. Thus an EER of 29.7% can be achieved for the wavelet based approach in fifth iteration. The fusion of multiple iteration levels does not improve the result significantly.

The employment of machine learning algorithms effects an enormous improvement of the detection performance of the algorithms. The EER drops to 5.0% on the development set. Contrary to the basic approach, the employment of a fusion of the wavelet transformation iterations is much more successful.

In general, the performance of the tested algorithms is lower on other data sets. On the evaluation set the best EER increases by 2.1 percent points, for the ASVspoof data the EER at least doubles for most algorithms. For the best algorithm the performance increases by 6.7 percent points to 11.7%. The character of the DET plot remains the same for different data sets. The increasing performance of the fusion less wavelet approach in fifth iteration in combination with a SVM on the ASVspoof data yields, that further research on the tuning of features and machine learning algorithms may improve the detection performance.

The increase of the amount of training data for the machine learning algorithms results in no significant improvement. Especially, the performance for the evaluation set drops, which is caused by overfitting of the machine learning algorithms. The best EER achieved raises by 4.2 percent points to 11.3%. The result of the fusion less wavelet approach in fifth iteration in combination with a SVM is remarkable. The EER drops again by 0.4 percent points, outperforming the other approaches.

In general, features of the Fourier spectrum and wavelet transformation with fusion perform good on the development and evaluation set. On the ASVspoof dataset, the state-of-the-art algorithm presented in 3.6.4 excels the presented approaches by 2.6 percent points, yielding an EER of 8.5%

The proposed algorithm performing best on the ASVspoof dataset yields an EER of 11.1%. Table 10 compares the presented algorithms with three approaches proposed at the spoofing challenge 2015 yielding the best unit-selection detection performance. The combination of CFCCIF and MFCC yields a slighly better performance, which is likely caused by the utilization of the phase-considering IF. Pure

phase-based features, i.e. CNPCC, which are eligible for detecting synthesis and voice conversion attacks, are outperformed by the algorithms proposed in this thesis, although the SVM is not trained on ASVspoof data.

Table 10: Comparison of proposed countermeasures to algorithms introduced in ASVspoof

Features	**EER**
DWT-fusion+FFT+SVM	11.7%
DWT-5+FFT+SVM	11.7%
DWT-fusion+FFT+SVM+full-training-set	13.8%
DWT-5+FFT+SVM+full-training-set	11.1%
CFCCIF+MFCC [6]	8.49%
High Dimensional Magnitude and Phase [4]	26.1%
CNPCC [73]	26.39%

4.7 FUTURE WORK

In this thesis, novel approaches for unit-selection detection are proposed. The presented feature extraction methods are able to distinguish bona fide and attack samples. In order to improve the performance of the countermeasures further research is recommended.

The proposed methods utilize a Fourier transformation for extracting a fix-dimension feature vector. The utilization of further frequency analysing techniques, like the early discarded STFT can be examined. As introduced in section 4.1.4, an edge detection or pattern recognition on the spectrogram promises the capability of creating a discriminative feature vector for bona fide and attack samples. A fusion of edge detection and Fourier-based features can be considered as well.

In this thesis, the magnitude of the transformed signal. Further improvements are prospective, if a feature is created with the complex-valued result of the Fourier transformation, as current research shows the importance of the consideration of phase-information in the field of PAD.

The machine learning algorithms utilized in this thesis are configured based on experience values. Further examination of different configurations of the machine learning algorithms may improve the discrimination performance on the developed feature vectors. Also the examination of further machine learning techniques, i.e. Deep Neuronal Networks (DNNs) or Mult Layer Perceptrons (MLPs), may improve the detection performance.

Further, the creation of unit-selection attacks requires additional research. For example, the decisive parameters for the quality of unit-selection attacks are not determined yet. Also there is no standardised metric for the performance estimation of unit-selection attacks. A further topic to examine is the dependence between quality of unit-selection attacks and the detection performance of PAD subsystems.

5

CONCLUSION

This thesis examines PAD for unit-selection attack. The focus is set on frequency based feature vectors extracted from unfiltered speech signals. Multiple detection algorithms are proposed, evaluated and improved. The evaluation is realised according to ISO/IEC CD2 30107-3 utilizing the ASVspoof database.
In general, biometric systems are vulnerable against attacks. Due to the advanced technology for speech synthesis, SIV systems are threatened in particular. This thesis examines the PAD performance of standard SIV systems on unit-selection attacks. State-of-the-art SIV systems are not capable of distinguishing bona fide and attack samples, thus additional countermeasures are needed.
A basic examined PAD algorithms did not perform well. The results of the machine learning approach was good for the development set. The performance drop for the ASVspoof data requires further research regarding the universality of the features. The improvement of the fusion less wavelet approach shows, that further investigation on the feature vectors can be meaningful.
The spectrogram based approach, discarded after the first evaluation, could be improved by a wavelet transformation, detecting edges in the spectrogram. Another possibility of improving the features could be the consideration of phase informations. For length normalization the FFT of the features is calculated. In the presented algorithms, the magnitude was utilized. A feature vector preserving the phase promise to lead to a more accurate classification.
The approach utilising machine learning outperforms the basic algorithms by far. Further research on configuration and training of the GMM and SVM is expected to improve the performance of the classification. The GMM shows slightly better results for larger training sets.
The ASVspoof dataset provides a solid basis for the uniform evaluation of PAD systems. The algorithms developed in this thesis scored second on the dataset, benchmarked with the submissions to the spoofing challenge.
This thesis demonstrates, that PAD without the usage of speech-recognition related features like MFCCs and CFCCs is possible. Further research on the frequency analysis of unfiltered speech signals promises further improvement of the detection performance.

allophone Phonemes can be pronounced different, the possible pronunciations are called allophones. 43

attack potential Attribute of a biometric presentation attack expressing the effort expended in the preparation and execution of the attack in terms of elapsed time, expertise, knowledge about the capture device being attacked, window of opportunity and equipment, graded as Basic, Enhanced-Basic, Moderate, High, or Beyond High [15]. 32

attack type Set of presentation attacks distinguished either by a common recipe for creating the artefact including the parameters (e.g. regarding environment or interaction method) used in the biometric capture process or by a common method of mutilating, altering, or imitating a human biometric characteristic [15]. 31

biometric characteristic Biological and behavioural characteristic of an individual from which distinguishing, repeatable feature can be extracted for the purpose of biometric recognition [17]. 2, 6–9

biometric enrolment Act of creating and storing a biometric enrolment data record in accordance with an enrolment policy [17]. 6

biometric feature Numbers or labels extracted from biometric samples and used for comparison [17]. 2, 5–8

biometric identification Process of searching against a biometric enrolment database to find and return the biometric reference identifier(s) attributable to a single individual [17]. 2, 6, 7, 30

biometric probe Biometric sample or biometric feature set input to an algorithm for use as the subject of biometric comparison to a biometric reference(s) [17]. 6, 7, 32, 61

biometric recognition Automated recognition of individuals based on their biological and behavioural characteristics [17]. 2, 8

biometric reference One or more stored biometric samples, biometric templates or biometric models attributed to a biometric data subject and used as the object for biometric comparison [17]. 6, 7, 32, 61

biometric sample Analog or digital representation of biometric characteristics prior to biometric feature extraction [17]. 5–9, 11, 15, 82

biometric verification Process of confirming a biometric claim through biometric comparison [17]. 2, 6, 7, 9, 30

bona fide presentation Interaction of the biometric capture subject and the biometric data capture subsystem in the fashion intended by the policy of the biometric system [15]. 3, 46, 48, 50, 51, 56, 58, 60–62, 64–66, 70, 71, 73, 78, 80

challenge response A type of protocol, characterized by one entity sending a challenge to another entity. The second entity must respond with the appropriate answer. 33, 34

concealment The attempt to be not identified by a biometric system. 30

diphone A diphone is a segment of speech that starts at the middle of one phone and extends to the middle of the next phone. The cut points are located in the middle of the phonemes, in the acoustic most stable region [32]. 14, 34, 44, 59

formant f0 Formants are frequency-ranges with high gain, caused by resonances. The lowest formant is referred to as f0. 13, 33, 34, 43, 44

overfitting In the field of machine learning, overfitting refers to a too excessive trained on a training population, so the accuracy of the model for further populations is reduced. 73, 78

phoneme A phoneme is the smallest sound unit, that is needed to distinguish two words. 12–14, 43, 44, 52

prosody The rhythm and pattern of sounds of poetry and language. 14

replay attack The speech of the attacked person is recorded and replayed in front of the microphone of the SIV [17]. 33, 35

signal energy The energy of a signal is the sum over the absolute squares of its time-domain samples. 18

signal power The power of a signal is the sum of the absolute squares of its time-domain samples divided by the signal length. 14

subversive biometric capture subject Biometric capture subject who attempts to subvert the correct and intended policy of the biometric capture subsystem [17]. 30

subversive users user of a biometric system who attempts to subvert the correct and intended system policy [17]. 30

syllable A syllable is a concatenation of phonemes, which can be vocalized at once. 13

unit-selection TTS-System based on the concatenation of speech samples. Regarding target cost and join cost the suitable units are selected during the synthesis process. 2–4, 14, 34, 35, 43–46, 50–52, 54–61, 64, 65, 69, 78, 80, 81

voice transcription A textual representation of a corresponding speech sample. A file containing a transcription is referred to as transcription file. 43, 59, 60

BIBLIOGRAPHY

[1] International Organization for Standardization, *Information technology – Biometric performance testing and reporting – Part 4: Interoperability performance testing*, JTC 1/SC 37 ISO/IEC 19 795-4:2008, 2008.

[2] A. K. Jain, A. A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, jan 2004.

[3] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Biometrics*. Boston, MA: Springer US, 2008.

[4] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li, "Spoofing Speech Detection Using High Dimensional Magnitude and Phase Features: The NTU Approach for ASVspoof 2015 Challenge," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2052–2056.

[5] A. Miguel, A. Ortega, E. Lleida, J. Villalba, A. Miguel, A. Ortega, and E. Lleida, "Spoofing Detection with DNN and One-class SVM for the ASVspoof 2015 Challenge," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2067–2071.

[6] T. B. Patel and H. A. Patil, "Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2062–2066.

[7] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, jan 2010.

[8] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, may 2002, pp. IV–4072–IV–4075.

[9] C. Busch and C. Sousedik, "Presentation attack detection methods for fingerprint recognition systems: a survey," *IET Biometrics*, vol. 3, no. 4, pp. 219–233, dec 2014.

[10] G. Pan, Z. Wu, and L. Su, "Liveness Detection for Face Recognition," in *Recent Advances in Face Recognition*. InTech, jun 2008, no. December, ch. 9, p. 236.

[11] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, A. Sizov, C. Hanilc, A. Sizov, and U. Kingdom, "ASVspoof 2015: the First Automatic Speaker Verification Spoofing and Countermeasures Challenge," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2037–2041.

[12] J. Villalba and E. Lleida, "Speaker Verification Performance Degradation against Spoofing and Tampering Attacks," in *Proc. Fala Workshop 2010*, 2010, pp. 131–134.

[13] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the Vulnerability of Speaker Verification to Synthetic Speech," in *Proc. IEEE Speaker and Language Recognition Workshop (Odyssey)*, 2010, pp. 151–158.

[14] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, feb 2015.

[15] International Organization for Standardization, *Information Technology – Biometric presentation attack detection – Part 3: Testing and reporting*, JTC 1/SC 37 ISO/IEC CD2 30 107-3:2016, 2016.

[16] ——, *Information technology – Biometric performance testing and reporting – Part 1: Principles and framework*, JTC 1/SC 37 ISO/IEC 19 795-1:2006, 2006.

[17] ——, *Information technology – Vocabulary – Part 37: Biometrics*, JTC 1/SC 37 ISO/IEC 2382-37:2012, 2012.

[18] S. Prabhakar, S. Pankanti, and A. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security & Privacy Magazine*, vol. 1, no. 2, pp. 33–42, mar 2003.

[19] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. London: Springer London, 2009.

[20] S. Z. Li and A. K. Jain, Eds., *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2015.

[21] G. Doddington, "Speaker recognition: Identifying people by their voices," in *IEEE 73.11*, vol. 73, no. 11, 1985, pp. 1651–1664.

[22] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Qin Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and Bing Xiang, "The SuperSID project: exploiting high-level information for high-accuracy speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP).*, vol. 4, no. 1. IEEE, 2003, pp. IV–784–7.

[23] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3-4, pp. 455–472, jul 2005.

[24] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[25] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Tech. Rep., 2006.

[26] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, may 2011.

[27] D. Garcia-Romero and C. Espy-Wilson, "Analysis of I-vector Length Normalization in Speaker Recognition Systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[28] D. Meuwly and A. Drygajlo, "Forensic Speaker Recognition Based on a Bayesian Framework and Gaussian Mixture Modelling (GMM)," in *Proc. 2001: A Speaker Odyssey The Speaker Recognition Workshop*, 2001.

[29] J. Schroeter, "Basic Principles of Speech Synthesis," in *Springer Handbook of Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, no. 1, pp. 413–428.

[30] Oxford University Press. (2015) Dictionary, Oxford English. "OED.". Accessed: 2016-01-24. [Online]. Available: www.oed.com

[31] J. P. H. Van Santen, "Combinatorial issues in text-to-speech synthesis," *Proc. 5th European Conference on Speech Communication and Technology*, pp. 4–7, 1997.

[32] J. Olive, "Rule synthesis of speech from dyadic units," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, vol. 2. Institute of Electrical and Electronics Engineers, 1977, pp. 568–570.

[33] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis." in *Proc. Eurospeech*, Madrid, 1995, pp. 581–584.

[34] J. Daugman, "How Iris Recognition Works," in *The Essential Guide to Image Processing*. Elsevier, 2009, vol. 14, no. 1, pp. 715–739.

[35] C. Bagwell, *SoX - Sound eXchange, the Swiss Army knife of audio manipulation*, 2013, accessed: 2015-11-28. [Online]. Available: http://sox.sourceforge.net/sox.html

[36] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, vol. 1, no. 1. IEEE, may 2013, pp. 7229–7233.

[37] S. S. Meduri and R. Ananth, "A Survey and Evaluation of Voice Activity Detection Algorithms," Blekinge Tekniska Högskola, Tech. Rep., 2011.

[38] S. M. Alessio, *Digital Signal Processing and Spectral Analysis for Scientists*, ser. Signals and Communication Technology. Cham: Springer International Publishing, 2016.

[39] H. Sun, B. Ma, and H. Li, "Frame selection of interview channel for NIST speaker recognition evaluation," in *Proc. 7th International Symposium on Chinese Spoken Language Processing, (ISCSLP)*. IEEE, nov 2010, pp. 305–308.

[40] W. Böge and W. Plaßmann, *Vieweg Handbuch Elektrotechnik*, W. Böge and W. Plaßmann, Eds. Wiesbaden: Vieweg+Teubner, 2007, vol. 53, no. 9.

[41] L. Papula, *Mathematik für Ingenieure und Naturwissenschaftler Band 2*. Wiesbaden: Vieweg+Teubner Verlag, 2011, vol. 53, no. 9.

[42] A. Mertins, *Signaltheorie*. Wiesbaden: Springer Fachmedien Wiesbaden, 2013.

[43] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, nov 1946.

[44] G. Strang, "Wavelet transforms versus Fourier transforms," *Bulletin of the American Mathematical Society*, vol. 28, no. 2, pp. 288–306, apr 1993.

[45] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, jul 1989.

[46] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, jan 1992, vol. 61.

[47] G. Tzanetakis, G. Essl, and P. Cook, "Audio Analysis using the Discrete Wavelet Transform," in *Proc. Conference in Acoustics and Music Theory Applications, (WSES)*, 2001.

[48] E. Alpaydin, *Machine Learning*. Oldenbourg Wissenschaftsverlag GmbH, 2008.

[49] C.-M. Huang, Y.-J. Lee, D. K. Lin, and S.-Y. Huang, "Model selection for support vector machines via uniform design," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 335–346, sep 2007.

[50] N. Ayat, M. Cheriet, and C. Suen, "Automatic model selection for the optimization of SVM kernels," *Pattern Recognition*, vol. 38, no. 10, pp. 1733–1745, oct 2005.

[51] S. J. D. Prince, *Computer Vision*. Cambridge: Cambridge University Press, 2012.

[52] International Organization for Standardization, *Information Technology – Biometrics presentation attack detection – Part 1: Framework*, JTC 1/SC 37 ISO/IEC 30 107-1:2015, 2015.

[53] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. A. Reynolds, "Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation," *National Institut of Standards and Technology Gaithersburg*, pp. 1–4, 1998.

[54] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, T. Leino, and A.-M. Laukkanen, "I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry," in *Proc. 14th Annual Conference of the International Speech Communication Association, (ISCA)*, 2013, pp. 930–934.

[55] C. Yang, G. Hammouri, and B. Sunar, "Voice Passwords Revisited," in *SECRYPT*, 2012, pp. 163–171.

[56] J. Lindberg and M. Blomberg, "Vulnerability in speaker verification-a study of technical impostor techniques." *Proc. Eurospeech*, vol. 3, no. MARCH 2001, pp. 1211–1214, 1999.

[57] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE - Trans. Inf. Syst*, pp. 816–824, 2007.

[58] L.-W. Chen, W. Guo, and L.-R. Dai, "Speaker Verification against Synthetic Speech," in *Proc. 7th International Symposium on Chinese Spoken Language Processing, (ISCSLP)*. IEEE, nov 2010, pp. 309–312.

[59] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, and D. Erro, "The AHOLAB RPS SSD Spoofing Challenge 2015 Submission," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2042–2046.

[60] L. Wang, Y. Yoshida, Y. Kawakami, and S. Nakagawa, "Relative Phase Information for Detecting Human Speech and Spoofed Speech," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2092–2096.

[61] J. J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *Proc. Carnahan Conference on Security Technology*. IEEE, oct 2011, pp. 1–8.

[62] P. L. De Leon, B. Stewart, and J. Yamagishi, "Synthetic Speech Discrimination using Pitch Pattern Statistics Derived from Image Analysis." in *Proc. Interspeech*, 2012, pp. 370–373.

[63] A. Ogihara, "Discrimination Method of Synthetic Speech Using Pitch Frequency against Synthetic Speech Falsification," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 1, pp. 280–286, jan 2005.

[64] DFKI GmbH, "Unit selection voice creation and explanation on Individual Voice Import Components," 2014, accessed: 2016-02-22. [Online]. Available: https://github.com/marytts/marytts/wiki/UnitSelectionVoiceCreation

[65] A. W. Black, "Perfect synthesis for all of the people all of the time," in *Proc. IEEE Workshop on Speech Synthesis*. IEEE, 2002, pp. 167–170.

[66] M. Schröder and A. Hunecke, "Creating German Unit Selection Voices for the MARY TTS Platform from the BITS Corpora," in *Proc. 6th ISCA Tutorial and Research Workshop on Speech Synthesis (SSW6)*, 2007, pp. 95–100.

[67] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu, "Robust Deep Feature for Spoofing Detection — The SJTU System for ASVspoof 2015 Challenge," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2097–2101.

[68] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "ASVspoof 2015: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 1–5.

[69] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A Comparison of Features for Synthetic Speech Detection," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2087–2091.

[70] A. Janicki, "Spoofing Countermeasure Based on Analysis of Linear Prediction Error," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2077–2081.

[71] Y. Liu, Y. Tian, L. He, J. Liu, and M. T. Johnson, "Simultaneous Utilization of Spectral Magnitude and Phase Information to Extract Supervectors for Speaker Verification Anti-Spoofing," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2082–2086.

[72] C. Hanilçi, T. Kinnunen, M. M. Sahidullah, A. Sizov, C. Hanilci, T. Kinnunen, M. M. Sahidullah, and A. Sizov, "Classifiers for Synthetic Speech Detection: A Comparison," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2057–2061.

[73] J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis, "Development of CRIM System for the Automatic Speaker Verification Spoofing and Countermeasures Challenge 2015," in *Proc. 16th Annual Conference of the International Speech Communication Association, (ISCA)*, 2015, pp. 2072–2076.

[74] Q. Li and Y. Huang, "An Auditory-Based Feature Extraction Algorithm for Robust Speaker Identification under Mismatched Conditions," *IEEE, Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1791–1801, 2011.

[75] Q. Li, "An auditory-based transfrom for audio signal processing," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, no. 1. IEEE, oct 2009, pp. 181–184.

[76] Z. Wu, A. Khodabakhsh, C. Demiroglu, J. Yamagishi, D. Saito, T. Toda, and S. King, "SAS: A speaker verification spoofing database containing diverse attacks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, no. 301. IEEE, apr 2015, pp. 4440–4444.

[77] O. Glembek, L. Burget, and P. Matějka, "Voice Biometry Standard - Draft," Biometry Standardization Initiative, Tech. Rep., 2015.

[78] D. Schnelle-Walka, S. Radeck-arneth, C. Biemann, and S. Radomski, "An Open Source Corpus and Recording Software for Distant Speech Recogni- tion with the Microsoft Kinect," in *Proc. 11. ITG Fachtagung Sprachkommunikation*, 2014, pp. 1–4.

[79] E. Khoury, T. Kinnunen, A. Sizov, Z. Wu, and S. Marcel, "Introducing I-vectors for joint anti-spoofing and speaker verification," in *Proc. 15th Annual Conference of the International Speech Communication Association, (ISCA)*, no. September, 2014, pp. 61–65.

[80] L. Debnath and F. A. Shah, *Wavelet Transforms and Their Applications*. Boston, MA: Birkh{ä}user Boston, 2015, vol. 56, no. 4.

[81] S. Hanov, "Wavelets and Edge Detection CS698 Final Project," Tech. Rep., 2006.

[82] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, jun 2000.