Secure, efficient automatic speaker verification for embedded applications

Giacomo Valenti

PHD THESIS
SORBONNE UNIVERSITÉ
Thèse de doctorat (CIFRE)

École doctorale ED 130
Informatique, Telecommunications et Electronique de Paris

EURECOM - Sécurité numérique

Thesis director : prof. Nicholas Evans

presented by

Giacomo VALENTI

Thesis title :

Secure, efficient automatic speaker verification for embedded applications

defense scheduled on 4th March 2019

before a committee composed of :

Prof.	Magne JOHNSEN	Rapporteur
Prof.	Tomi KINNUNEN	Rapporteur
Prof.	Marc DACIER	Examiner
M.	Nicolas OBIN	Examiner
Mme	Florence TRESSOLS	Examiner
M.	Adrien DANIEL	Examiner
M.	Laurent PILATI	Examiner, industrial supervisor
Prof.	Nicholas EVANS	Thesis director

Abstract

This industrial CIFRE PhD thesis addresses automatic speaker verification (ASV) issues in the context of embedded applications. The first part of this thesis focuses on more traditional problems and topics, which are introduced with a specific literature review. The first work investigates the minimum enrolment data requirements for a practical, text-dependent short-utterance ASV system. Results on the RSR2015 database and protocols indicate that the need for up to 97% enrolment data can be eliminated with only a negligible impact on performance.

Contributions in the first part of the thesis consist in a statistical analysis onto the aforementioned RSR corpus. The objective is to isolate text-dependent factors and prove they are consistent across different sets of speakers. Experiments suggest that, for very short utterances, the influence of a specific text content on the system performance can be considered a speaker-independent factor and is named *spoken password strength*. If the user could be made aware of the strength of the chosen password, ASV reliability could be improved with the judicious choice of a more secure password over a less secure one.

The second part of the thesis focuses on neural network-based approaches and is accompanied by a second, specific literature review. While it was clear from the beginning of the thesis that neural networks and deep learning were becoming state-of-the-art in several machine learning domains, their use for embedded solutions was hindered by their complexity.

Contributions described in the second part of the thesis comprise blue-sky, experimental research which tackles the substitution of hand-crafted, traditional speaker features in favour of operating directly upon the raw audio waveform and the search for optimal network architectures and weights by means of genetic algorithms. This work is the most fundamental contribution of this thesis: neuro-evolved network structures consisting of only a few hundred connections which are able to learn from the raw audio input. While the approach is undoubtedly still in its infancy, it shows promising results for experiments carried out for text-independent speaker verification (including a subset of the NIST 2016 SRE data) and anti-spoofing (on the official ASVspoof 2017 protocols).

Acknowledgements

This thesis was carried out at EURECOM and NXP Semiconductors and was entirely funded by NXP. During this 3-year long journey, lasting from July 2015 to August 2018, I was not alone.

I would like to thank my academic advisor, Prof. Nicholas Evans for his relentless support not only during the time of this thesis, but also during the preceding 5 months, which he allowed me to spend at EURECOM, in order to familiarise with speaker verification. My gratitude also goes to Dr. Adrien Daniel, who has been my industrial supervisor for the most part of my thesis. He is responsible for pushing me to explore new horizons of research; our active collaboration led to the most fundamental contributions of this thesis.

Since I had a few jobs before working at NXP Semiconductors, I can sincerely say that the NXP working environment, especially at the human level, is the best I ever experienced. I am very much grateful to my boss and current industrial supervisor, Laurent Pilati, for letting me do research with a great degree of freedom, something which I am told is not to be taken for granted in industrial PhDs. A special thank goes to my former classmate, former colleague and current friend, Daniele Battaglino, who introduced me to my academic supervisor and to NXP at a time when I was really dissatisfied with my professional life.

Very warm thanks go to my soulmate Francesca, her "never-settle" attitude is what pushed me to leave a stable (albeit boring) job for a definitely more adventurous experience. Among the people who encouraged me and always supported me I cannot avoid to mention my parents, whom I deeply thank for teaching me the importance of education since my first day of school.

I would also like to express my gratitude to my fellow PhD students and post-docs in EURECOM for the nice atmosphere every time I "dropped by", with special thanks to Héctor for his patience and kindness. Last, at the risk of repeating myself with regard to my master thesis' acknowledgements, I want to thank friends in my home town. They have exactly nothing to do with audio-related research or data science, but they know the true meaning of "decompress".

Giacomo Valenti

Contents

Abstract iii

Acknowledgements v

I Introduction 1
- I.1 Speaker recognition terminology . . . 2
- I.2 Issues with current ASV status . . . 3
- I.3 Contributions and publications . . . 4
- I.4 Thesis structure . . . 7
  - I.4.1 Part A . . . 7
  - I.4.2 Part B . . . 8

II A review of traditional speaker verification approaches 11
- II.1 Speech as a biometric . . . 11
- II.2 Front-end: speaker features . . . 12
  - II.2.1 Short-term features . . . 13
  - II.2.2 Longer-term features . . . 14
- II.3 Back end: models and classifiers . . . 15
  - II.3.1 Gaussian Mixture Models . . . 15
  - II.3.2 GMM-UBM . . . 17
  - II.3.3 Hidden Markov Models . . . 17
  - II.3.4 Towards i-vectors . . . 19
  - II.3.5 The HiLAM system . . . 20
- II.4 Performance Metrics . . . 21
  - II.4.1 Receiver Operating Characteristic (ROC) . . . 21
  - II.4.2 Equal Error Rate (EER) . . . 23
  - II.4.3 Score normalisation . . . 23
- II.5 Challenges and Databases . . . 23
  - II.5.1 NIST Speaker Recognition Evaluations . . . 23
    - II.5.1.a The early years . . . 24
    - II.5.1.b Broader scope and higher dimensionality . . . 24
    - II.5.1.c Bi-annual big data challenges . . . 24
    - II.5.1.d SRE16 . . . 25
  - II.5.2 RSR2015 corpus . . . 26

II.5.2.a Training . . . 26
II.5.2.b Testing . . . 27
II.6 Summary . . . 27

III Simplified HiLAM **29**
III.1 HiLAM baseline implementation . . . 30
III.1.1 Preprocessing and feature extraction . . . 30
III.1.2 GMM optimisation . . . 31
III.1.3 Relevance factor optimisation . . . 31
III.1.4 Baseline performance . . . 31
III.2 Protocols . . . 32
III.3 Simplified HiLAM . . . 33
III.3.1 Middle-layer training reduction . . . 33
III.3.2 Middle layer removal . . . 34
III.4 Evaluation Results . . . 35
III.5 The Matlab demo . . . 37
III.6 Conclusions . . . 38

IV Spoken password strength **39**
IV.1 The concept of spoken password strength . . . 39
IV.2 Preliminary observations . . . 40
IV.2.1 The text-dependent shift . . . 40
IV.2.2 The text-dependent overlap . . . 40
IV.3 Database and protocols . . . 42
IV.4 Statistical analysis . . . 42
IV.4.1 Variable strength command groups . . . 42
IV.4.2 Sampling distribution of the EER . . . 43
IV.4.3 Isolating the influence of overlap . . . 43
IV.5 Results interpretation . . . 45
IV.6 Conclusions . . . 46

V A review of deep learning speaker verification approaches **47**
V.1 Neural networks and deep learning . . . 47
V.1.1 Deep Belief Networks . . . 48
V.1.2 Deep Auto-encoders . . . 49
V.1.3 Convolutional Neural Networks . . . 50
V.1.4 Long short-term Memory Recurrent Neural Networks . . . 50
V.2 Deep learning in ASV . . . 51
V.2.1 Feature extraction . . . 52
V.2.2 Applications to i-vector frameworks . . . 52
V.2.3 Back-ends and classifiers . . . 52
V.3 End-to-end . . . 53
V.3.1 Middle-level representations VS raw audio . . . 53
V.3.2 Fixed topologies . . . 54
V.4 Summary . . . 55

VI Augmenting topologies applied to ASV **57**
VI.1 Evolutionary strategies . . . 58
VI.1.1 TWEANNs . . . 58
VI.1.2 NEAT . . . 60
VI.2 Application to raw audio classification . . . 62
VI.3 Truly end-to-end automatic speaker verification . . . 65
VI.3.1 Fitness function . . . 65
VI.3.2 Mini-batching . . . 66
VI.3.3 Training . . . 66
VI.3.4 Network selection for evaluation . . . 68
VI.4 Experiments . . . 68
VI.4.1 Baseline systems . . . 69
VI.4.2 NXP database and experimental protocols . . . 70
VI.4.3 End-to-end system: augmentation and generalisation . . . 71
VI.5 Further experiments:
End-to-end system on NIST SRE16 data . . . 72
VI.6 Conclusions . . . 76

VII Augmenting topologies applied to anti-spoofing **79**
VII.1 A brief overview of anti-spoofing . . . 80
VII.2 NEAT setup . . . 81
VII.2.1 Ease of classification . . . 81
VII.2.2 Training . . . 83
VII.2.3 Testing . . . 84
VII.3 Experimental setup . . . 84
VII.3.1 Database, protocol and metric . . . 84
VII.3.2 Baseline systems . . . 85
VII.3.3 End-to-end anti-spoofing . . . 86
VII.4 Experimental results . . . 87
VII.4.1 Evolutionary behaviour . . . 87
VII.4.2 Spoofing detection performance . . . 87
VII.5 Conclusions . . . 88

VIII Conclusions **91**
VIII.1 From the laboratory into the wild . . . 91
VIII.2 Not all sentences are created equal . . . 93
VIII.3 Truly end-to-end ASV . . . 95
VIII.4 Truly end-to-end anti-spoofing . . . 96
VIII.5 Closing thoughts and future work . . . 99

Appendix A Published work **101**

Bibliography **141**

Chapter I

Introduction

The last decade has witnessed tremendous progress and growing interest in voice biometrics, including the focus of several significant industrial players such as Google, Amazon and Facebook. Smart assistants are now a reality thanks to speech being one of the most convenient and natural biometric traits to collect, requiring little to no human-machine interaction, as a result of almost all smart devices being equipped with at least one microphone.

Speech recognition is a standard feature in the operating system of any currently manufactured personal computer, smartphone, television and middle-to-high range car. *Speaker verification* is used increasingly by online banking services to authenticate account holder. This is the age of automatic speech/voice technology. The Internet Of Things (IoT) paradigm, embraced by many companies, foresees a near future in which every piece of technology is connected and equipped with machine learning capabilities. In a context where all the user's domestic appliances, the front door lock, the car are connected and voice-enabled to respond to commands and answer questions, making them report exclusively to the target user(s) brings enormous value to IoT technology as a whole.

This industrial (CIFRE) PhD fits into this scenario as an academic collaboration between EURECOM and NXP Semiconductors bringing speaker recognition to practical, real use-case scenarios, namely those including smart devices often characterised by low computation/memory resources and low consumption requirements. Speaker recognition is way less common in smart devices than speech recognition because it is more difficult to integrate it seamlessly: enrolling a speaker requires collaboration from the end user to collect his/her voice data without feeling too forced. Since the resulting speaker model is subject to privacy issues, the need to move as much processing as possible on the device itself is a key aspect, which is also beneficial for security. This calls for user-friendly, small-footprint secure speaker recognition, which puts many current approaches out of the picture, as they often rely on large computation and storage resources.

NXP is the 5th largest semiconductor company in the world, with more than 30000 employees and 9000 patent families, founded in 2006 by the historical Dutch electronics giant Philips. NXP customers are among the most recognised brands, including Apple, Huawei, Hyundai, Panasonic and Samsung. While the primary focus and business of the company is still hardware manufacturing, in recent years the trend of selling software bundled with hardware, to provide a complete solution to clients, has proven increasingly profitable.

For what concerns audio-related software, NXP is in a process of expanding its signal processing portfolio with machine learning approaches which can enhance already existing solutions and, in that sense, bring speech and speaker recognition capabilities to their software library. Activity in machine learning research development was already growing at the start of this PhD, albeit not in speaker recognition.

Albeit an industrial PhD, the room for research and experimentation that was left to the author during the PhD was considerable. None of the contributions of this thesis were actually tied to a commercial product, but the author feels safe to assume that, from the company perspective, it was a successful collaboration because it was renewed and he is currently employed full-time in NXP as a research engineer, continuing some of the work reported in this thesis.

I.1 Speaker recognition terminology

The focus of this thesis is speaker authentication, held closely aligned to the general topic of of speaker recognition. This section serves to clarify some of the terminology used in this thesis in the case the reader is not familiar with the research field. Speaker recognition is intended as the broad field that encompasses all recognition tasks based on voice biometrics, not to be confused with *automatic speech recognition* (ASR), whose aim is to recognise *what* is being said, not *who* said it.

The field is broken down into sub-fields which differ in purpose and objectives:

- **Speaker identification**

  Speaker identification is a closed-set classification task. During the training phase, a system learns to distinguish between N speakers, each one representing a class. At test time, trial utterances can only belong to one of the N speakers and the identification system assigns the utterance to the closest-matching class.

- **Speaker diarization**

  Speaker diarization is a segmentation and clustering task. Its purpose is to identify, in recordings involving multiple speakers, those intervals in which each speaker is active. Segmentation consists in splitting the recording in homogeneous speaker segments, *i.e* supposed speaker changes, points or turns. Clustering aims to assign each segment to the corresponding speaker.

- **Speaker verification**

  Automatic speaker verification (ASV) is a binary classification task. While ASV itself can be broken down into sub-categories, *i.e.* surveillance, in this thesis the term ASV is used interchangeably with "speaker authentication". The difference between authentication and surveillance stems upon the notion of cooperation.

While speaker *identification* and *diarization* operate in closed-set scenarios where the number of speakers to identify or separate is finite, speaker verification faces the challenges of identifying what makes a target speaker different from —literally— anyone else. A speaker verification scenario will always be open-set, and modelling the alternative

hypothesis (all potential impostors) is as important as modelling the target speaker. Verification is therefore the task that truly makes use of speech as a biometric. This thesis focuses solely on automatic speaker verification.

The modalities of speaker verification give birth to further sub-categories which prescind all adopted technologies and chosen approaches, but are nevertheless relevant, at the highest level, to all the blocks of the toolchain. The modality choice influences both the way speaker data is collected and how tests are performed.

Text-dependent speaker verification can instead rely on a few seconds of speech, hence its common association with short pass-phrases authentication scenarios. This is possible because the speech content could be limited to a single short sentence, thus requiring less data to train the model and only one utterance at test. The quantity of training data is the main focus of chapter III. This scenario can be expanded by having each user utter several sentences, and thus having multiple text-dependent models per speaker. With this configuration, text mismatch comes into the picture: four combinations are now possible, with the only true target trial being the one where both text and speaker match. If the set of sentences is shared across the whole dataset, results can be grouped per-sentence and give different levels of insights on the influence of text content on system performance. This aspect is explored in chapter IV.

Text-independent speaker verification implies modelling a speaker and testing without any constraint on the speech content. It is understandable that in such a context, a solid speaker model should be versatile enough to authenticate the speaker with any utterance at test time. This is why text-independent approaches usually require at least 30 seconds of speaker data in order to be reliable [1]. Contributions in chapters VI and VII adopt a text-independent approach.

Text-prompted speaker verification is a variation on text-dependent verification which incorporates some text-independent aspects. Included in this category are the scenarios where the speaker is asked, at test, to utter a specific sentence which the system was not necessarily trained for. This usually implies a text-independent speaker model that was phonetically structured [2], meaning that it incorporated a broad knowledge of how the target speaker utters specific phonetic patterns. This is usually achieved by segmenting the training data with some form of speech recogniser or utterance verification [3].

I.2 Issues with current ASV status

In many practical use-cases ASV technology is often seen by users as somewhat cumbersome and non-secure, making it not worth the effort. This is in contrast with the non-intrusive nature of using speech as an interface, which is the main reason behind the increasingly wider adoption of voice-driven smart assistants. Both Amazon Echo and Google Home assistants have some speaker recognition capabilities but, as of today, they are not enabled by default and their use is confined to automatic user preference setting

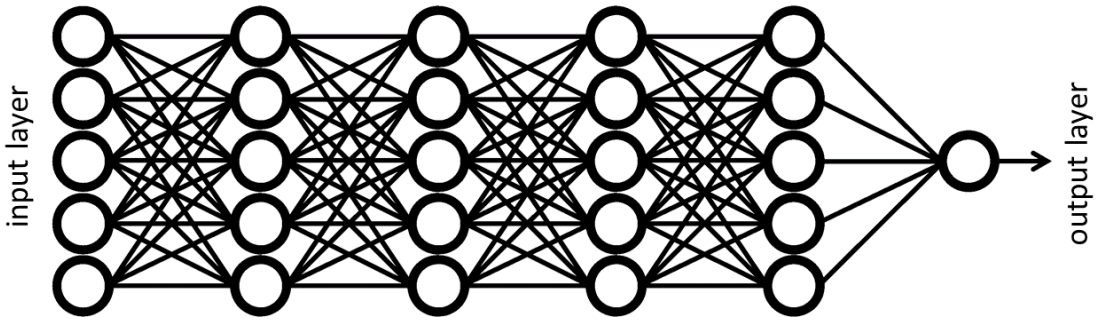


Fig. I.1 *An example structure of a deep neural network (DNN) with 26 units and 105 connections. In reality, state-of-the-art DNNs are at least 3 orders of magnitude more complex.*

or to lowering the false acceptance rate for wake-up word detection [1]. As a result, this feature is left mostly unused. If ASV was as mature as speech recognition on these kinds of devices, not only would its use be more publicised and widespread, but users would use their voice as a means to seamless authentication in a secured environment.

The road to reach this goal follows two complementary directions: *efficiency* and *security*. Efficiency refers to the perspectives of both the implementation and the end user: in order that ASV be deployed successfully on embedded devices, computationally- and memory-hungry approaches are incompatible; this is the case with complex deep neural network structures which require hundred of thousands of multiplications (see Fig. I.1) often on top of feature extraction. Also, to preserve convenience, an efficient system should require very little user speech in order to operate reliably.

The meaning of security is application-dependent: the level of security for a ASV-enabled parental control filter is different to that of an ASV system that is meant to control access to bank accounts. Nevertheless, any ASV functionality would be rendered useless if it could not distinguish between a real human and a recorded voice, a so-called *replay attack*. Fig. I.2 [2] illustrates how the error rate of an ASV system can be highly impacted when using replay attacks instead of genuine impostors. This, and other types of artificial voice attacks show the need for countermeasures, which in this domain are referred to as *anti-spoofing*.

I.3 Contributions and publications

During the 3-year period of research, all of the author's work was carried out while employed at NXP Semiconductors. Contributions thus exhibit an industrial flavour. They

[1] https://machinelearning.apple.com/2017/10/01/hey-siri.html

[2] Reproduced from a publicly available overview of the ASVspoof 2017 challenge: http://www.asvspoof.org/slides_ASVspoof2017_Interspeech.pdf

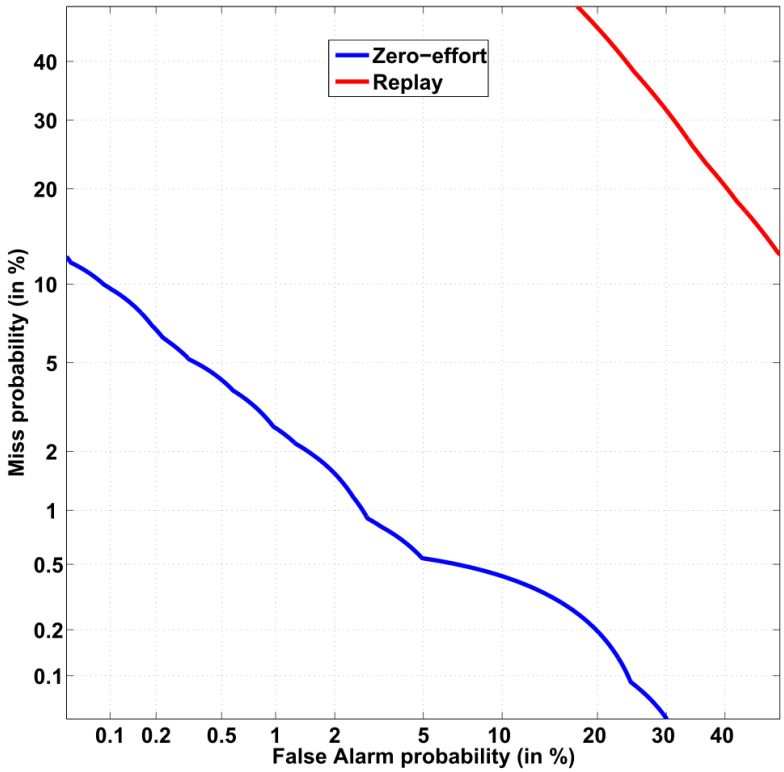


Fig. I.2 *An example of the impact of replay attacks versus genuine (zero-effort) impostors on the error rates of a GMM-UBM system* [2].

are listed below:

- **Improved usability for embedded ASV**

  This contribution aims to make an existing text-dependent system more user-friendly, to bridge the gap between academic research and end-user experience. With an (albeit modest) modification, the enrolment time for any phrase and user is reduced from approximately 5 minutes (which is deemed too much to ask to any end-user, even if it's required just once) to less than 10 seconds. The absence of notable drops in performances when reducing data to such an extent also calls into question the appropriateness of the training and testing protocol to assess performance. A MATLAB demo was implemented to show the practical increased usability of the system. The contribution is explained in Chapter III. Related work was published in:

  [4] Valenti, Giacomo; Daniel, Adrien; Evans, Nicholas
  "A simplified 2-layer text-dependent talker authentication system"
  143rd International AES Convention, 2017

- **The concept of spoken password strength**

  This contribution is a statistical proof that, in the context of very short commands (approximately 1 second of duration), speaker authentication performance is more

influenced by the text content than intra-speaker variance, defining the speaker-independent concept of *spoken password strength*.

With a thorough statistical analysis, it is proven that the authentication power of a spoken password is not just limited to a closed set of speakers, but it transfers well to an independent speaker set. This trait is thus assumed to be a characteristic of the spoken sentence, regardless of the speaker, and is termed *pass-phrase strength*. The related work is explained in Chapter IV and published in

[5] Valenti, Giacomo; Daniel, Adrien; Evans, Nicholas
"On the Influence of Text Content on Pass-Phrase Strength for Short-Duration Text-Dependent Automatic Speaker Authentication"
INTERSPEECH, 2016

This contribution was also the basis of a patent for an automated warning of weak spoken passwords, in the same fashion of written passwords in online account registration procedures:

[6] Valenti, Giacomo; Daniel, Adrien; Evans, Nicholas
"Spoken pass-phrase suitability determination"
Patent, 2016 (US 2018/0060557 A1)

- **Speaker Verification and Anti-spoofing on raw audio and topology-evolving neural networks**

These contributions relate to very experimental and fundamental research, despite the industrial flavour of this PhD. The potential of ASV and anti-spoofing solutions operating directly on the raw audio waveform is investigated. This work represents one of the first raw-audio-based ASV applications, as well as one of the first uses of Topology and Weight Evolving Neural Networks (TWEANNs) for aural tasks, an earlier example on *sound event detection* being [7].

The networks process input audio samples and output score samples at the same rate of the input, making for a **truly** end-to-end approach. A *gate* is applied to the output which, during training, learns to prune unreliable scores, akin to attention mechanisms. In the case of anti-spoofing, a new progress-rewarding fitness function is introduced and is shown to be beneficial for the task. This work was first reported in two publications:

[8] Valenti, Giacomo; Daniel, Adrien; Evans, Nicholas
"End-to-end automatic speaker verification with evolving recurrent neural networks"
ODYSSEY, 2018

[9] Valenti, Giacomo; Delgado, Héctor; Todisco, Massimiliano; Evans, Nicholas; Pilati, Laurent
"An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks"
ODYSSEY, 2018

I.4 Thesis structure

The structure of this thesis in divided in 2 parts (A and B), each of which is composed by a literature review followed by chapters which describe novel technical contributions. Chapters II, III and IV comprise part A, which is related to the fundamental treatment of traditional ASV and the first research activities and contributions of the author, specifically those involving short-utterance, text-dependent ASV.

Chapters V, VI and VII comprise part B, which focuses on more experimental and fundamental research concerning the use of raw-audio as input to neural networks with evolutionary topologies, in order to deliver a truly end-to-end pipeline. This raw-audio, non-fixed topology approach is applied to both ASV and anti-spoofing. It represents the very first work that applied evolutionary topologies and raw audio in either fields. The aspects of efficiency and security are tackled in all the contributions of this thesis, parts A and B (see Section I.3). The remainder of this thesis is organised as follows:

I.4.1 Part A

- **Chapter II - A review of traditional speaker verification approaches**

  The first part of this chapter introduces text-independent, text-dependent and text-prompted variations of speaker authentication and the challenges and peculiarities involved in using speech as a biometric. The traditional front-end and back-end approaches to ASV are then reviewed. "Traditional" infers the exclusion of neural network and deep learning approaches, they are treated separately in Chapter V.

  The front-end block of traditional ASV consists of preprocessing and feature extraction, with cepstral short-term features being the most widely used variant. Back-end approaches that are relevant to the contributions of this thesis are given a more detailed analysis, namely those based on Gaussian mixture models (GMM) and hidden Markov Models (HMM). More recent approaches like linear discriminat analysis (LDA), supervectors and i-vector approaches are also described. The HiLAM system (Hierarchical multi-Layer Acoustic Model), which is the basis system for contributions in Chapters III and IV, is then introduced.

  The second part of the chapter introduces the performance metrics used through the remainder of the thesis. A brief overview of the NIST Speaker Recognition Evaluations from 1996 to 2016 follows, with a focus on NIST SRE16 whose data and protocols were used in Chapter VI. The text-dependent RSR2015 corpus is described in detail, since it was used extensively used in experiments in Chapters III and IV.

- **Chapter III - Simplified HiLAM**

  This Chapter explains the re-implementation and simplification of the text-dependent, short-utterance ASV system HiLAM. The objective is to adapt a system made for offline experiments with stored data into a ready-to-use ASV system which requires just a few seconds to enrol a new speaker. The first part focuses on the re-implementation. Particular attention is given to the *relevance factor*, which

plays an influencing role in the *Maximum A Posteriori* (MAP) adaptation algorithm. Findings concerning the relevance factor lead to questioning the need for the text-independent modelling stage of the HiLAM pipeline.

The second part concerns the modification of the system by removing the text-independent model, simplifying it from a 3-layer to a 2-layer system which requires way less speaker data at enrolling time. Experiments are carried out to assess performance of the 3-layer versus the 2-layer variant as well as comparing to the original work. A MATLAB demo for the 2-layer system is also implemented, as a practical demonstration of the vastly reduced "cumbersome" aspect and increased usability.

- **Chapter IV - Spoken password strength**

  This chapter describes the study to prove the concept of spoken pass-phrase (or password) *strength*, which is inversely proportional to the probability of false positives and false negatives when a speaker text-dependent model relies on said password. The requirement to be able to quantify the strength of a spoken password is introduced: to be truly deemed stronger (more secure) than others, the strength concept has to be speaker-independent, ideally, as a property of the text content.

  Preliminary observations of text-dependent shifts in score distributions (supposedly linked to different pass-phrase strengths) are made on the short-commands protocols of the RSR2015 corpus. This shows how performance is greatly influenced by the text content when dealing with really short sentences.

  A statistical analysis is then performed to show that strength rankings made with respect to one set of speakers are coherent with a second, independent set of speakers. Interpretation of the results follows, therefore proving the concept of "universal" strength of a specific text content, independent of the speaker. A potential use case is envisioned in which the known a priori low strength of a user-chosen password is used to warn of the low security risk in choosing it.

I.4.2 Part B

During the first year and a half of this PhD, neural networks went from simply experiencing a resurgence, enhancing some confined blocks of ASV pipelines to being omnipresent, often making the difference and surpassing state-of-the art approaches. From a semiconductor company standpoint, neural networks are perceived as bulky and very resource-intensive, especially when *deep learning* comes into the picture. After investigating the most important deep learning approaches in their own literature review (Chapter V), instead of pursuing one of them, a radical departure in the direction of the highly-experimental domain of evolutionary topologies was taken, with their very small footprint being the main attraction. Hence the author decision to split this thesis into two main parts. Chapters VI and VII are centred around the explanation and application of such evolutionary topologies on raw-audio for ASV and anti-spoofing.

- **Chapter V - A review of deep learning speaker verification approaches**

  The first part of this chapter introduces the concept of *deep learning* and describes several types of Deep Neural Networks (DNN) which are used in ASV (and neighbouring research areas) such as Deep Belief Networks (DBN), Auto-encoders (AE), Convolutional Neural Networks (CNN) and Long Short-Term Memory Recurrent Neural Networks (LSTM RNNs).

  The second part of this chapter explains the ASV approaches in which DNNs are used to substitute a particular block of the traditional ASV pipeline. These approaches make use of DNNs as the front-end (*i.e.* as feature extractors) or as the back-end (*i.e.* as classifiers).

  The third and last part concerns *end-to-end* approaches. These apply deep learning to the whole ASV pipeline and all components are jointly optimised. The thesis then describes something of a misnomer as concerns end-to-end approaches in the literature since, despite the moniker of "end-to-end", current approaches when this work began tended to operate on spectral representations or even MFCCs, rather than the raw signal and employed fixed topologies instead of optimising the structure along with the weights.

- **Chapter VI - Augmenting topologies applied to ASV**

  Non-fixed architecture neural networks are explained in this chapter as belonging to the wider field of evolutionary strategies. Topology and Weight Evolving Neural Networks (TWEANNs) are introduced, as they use genetic algorithms to indeed optimise the structure along with the weights, and are at the basis of one of the first applications of the technology to raw audio, which is the contribution and focus of this chapter.

  A specific TWEANN approach known as NEAT (Neuro Evolution of Augmenting Topologies) is used as a basis and modified in order that it can be applied on raw audio signals and speaker verifications. Key aspects like the fitness function and the use of mini-batching are described, followed by non-conventional training and evaluation procedures.

  Experiments are performed on a proprietary NXP database and results are compared with several baselines consisting of a GMM-UBM system and two more conventional NN-based approaches. Particular attention is given to the learning process of the end-to-end system, which is monitored across several iterations of the evolutionary training algorithm: the overall decrease of the EER on training data is observed to correspond to decreases on the test-set data, exposing increasing generalisation capabilities. A first glimpse of "explicability" of neural network inner mechanics is observed by monitoring the gate unit behaviour on the output of tested networks. The end-to-end system is then also tested on a subset of the NIST SRE16 data. Results are compared to the ICMC IV PLDA system developed in EURECOM [10].

- **Chapter VII - Augmenting topologies applied to Anti-spoofing**

  This chapter describes the application of NEAT to the task of anti-spoofing. First, the concept of anti-spoofing is introduced because, even though it is strictly related to speaker authentication, as of today it is often treated as a separate task. Modifications to the NEAT-based system used for ASV are then explained, the most important being a new fitness function, named Ease Of Classification (EOC).

  The ASVspoof2017 database and protocols are then introduced along with the official baseline system. Experiments are then carried out and results for four different configurations of the end-to-end approach are examined. Particular attention is reserved to the improvements brought by the EOC fitness function combined with mini-batching.

- **Chapter VIII - Conclusions**

  This chapter concludes the thesis with a summary of all contributions and findings and a discussion of potential future work. It is highlighted how during the 3-year span the scope of the contributions went from a simple modification of an existing system to enhance usability, to detection of spoofed speech directly on raw-audio waveforms. Efficiency and security are key aspects of the author's work; they are both necessary for ASV to truly enter the everyday life, from both the user and the company points of view. The work in part A, focused on traditional ASV systems, demonstrated that text-dependent system may require very little speech to authenticate the user properly, but some sentences are more characterising than others for any speaker. The most important contributions are found in part B, where it is demonstrated that it is possible, however yet to be perfected, to perform ASV and anti-spoofing on the raw-audio waveform.

Chapter II

A review of traditional speaker verification approaches

This chapter explores the composing blocks that characterised decades of research in Automatic Speaker Verification, from feature extraction to modelling approaches to performance metrics (a high-level diagram of the traditional ASV pipeline is illustrated in Fig. II.1). The presentation focuses on the background knowledge that is most relevant to the contributions described in chapters III and IV of this thesis.

II.1 Speech as a biometric

The vast field of biometric science has the one goal of improving the reliability of technologies that discriminate and identify reliably different individuals by measuring their physical and behavioural traits. In many of them the latter is not considered because it is either non-conceivable (*i.e.* the behaviour of fingerprints) or the temporal dimension is often not necessary (*i.e.* face recognition). In most biometrics, the data that supposedly

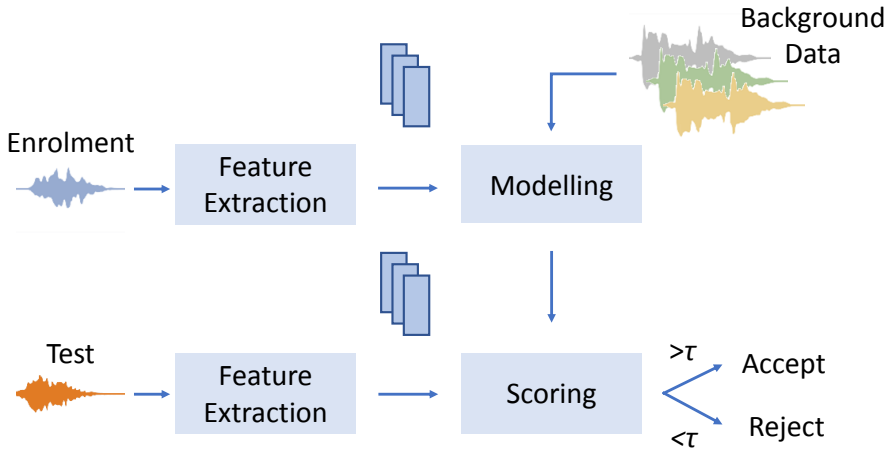


Fig. II.1 *Blocks of a traditional ASV pipeline, reproduced from [11]*

make an individual unique is captured, collected and studied directly on its natural domain: from finger prints to retinal scans, even DNA, the "serial number" of biometrics, is coded with a series of proteins.

Although the human vocal tract does exhibit individualising physical and behavioural aspects, most of the difficulty comes from the fact that neither the physical traits of the vocal tract, nor its movements are directly observable from an audio signal. The latter is purely a product, a manifestation of these factors. In analogy, the recognition of a speaker from their speech is akin to recognising a music band from their tracks (the instrumentation is the vocal tract, the playing style is the behaviour) rather than from a picture of the band members. Some questioned that speech is not enough unique to recognise an individual [12].

On the other hand, by being purely aural as opposed to many visual-based biometric sciences, speech presents itself as one of the least invasive biometrics to measure and easier to collect. Nevertheless, in more than 30 years of ASV research, the identifying cues have seldom been observed directly on the raw waveform representation, but through many hand-crafted methods of speaker feature extraction which involve several transformations of the original signal.

II.2 Front-end: speaker features

Recent trends in research, not strictly related to ASV, show the progressive abandoning of traditional features, and the will to leave the burden to machine learning. This aspect is of crucial focus for part of the work in this thesis (see chapters VI and VII) and will be explored later. The focus of this section is on traditional hand-crafted features, specifically those adopted in the work reported in Chapters III and IV. Regardless of the representation on which features are observed, the ideal speaker feature properties outlined by Nolan in 1983 [13] still apply. They are to:

1. show high between-speaker variability and low within-speaker variability;
2. be resistant to attempted disguise or mimicry;
3. have a high frequency of occurrence;
4. be robust in transmission;
5. be relatively easy to extract and measure;

Needless to say, the **ideal** feature does not exist. The speech signal also contains non-individualising information to which features should not be sensitive to, and different kinds of features can be extracted from the same data, depending on the application. Even leaving aside limits in computational power and data availability, there is no *passepartout* in feature selection [2].

The remainder of this section will go through the most widely used feature types, with their degree of success usually dependent upon a given application or context. In traditional ASV approaches, the feature extraction process remains the same for every speaker even though some studies show that, for example, some type of feature is better

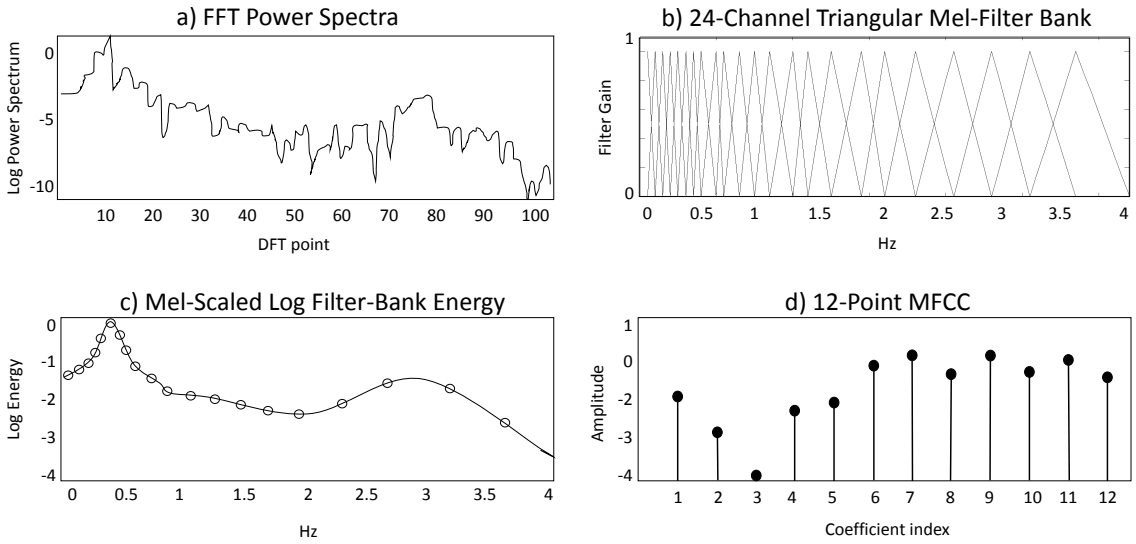


Fig. II.2 *MFCC features extraction steps, reproduced from [11]*

at discriminating female than male [14]. With the recent advent of *deep learning* applied to ASV, features are becoming less hand-crafted and more speaker-specific. This body of work is reviewed later in Chapter V.

II.2.1 Short-term features

Short-term cepstral features were initially engineered in the early 80s for speech recognition and with in mind the limitations of the modelling approaches of the time. They describe traits of the human vocal tract that are assumed to be stationary inside very short intervals. The low dimensionality of cepstral features brought not only less computational efforts for subsequent demanding processing, but was once a necessity since traditional statistical models could not handle high-dimensional data [2].

The first extraction step, common to all short-term approaches, is to segment the source audio into overlapped frames of fixed length (usually 10 to 30 milliseconds), each one is then processed independently. Since silence and pauses bring no relevant information to identify the speaker, silence frames (or samples, in case the process is done before framing) are discarded using *voice activity detection* (VAD) tools [11].

The most popular short-term features are cepstral in nature, with the most widely used being the MFCC (Mel Frequency Cepstral Coefficient) variant. Invented in 1980 [15] for speech recognition, MFCCs remained at the heart of speaker verification systems until very recently [1]. During this period ASV approaches evolved considerably (see Section II.5.1) but the vast majority of state-of-the-art solutions across decades adopted MFCC extraction (or cepstral features in general) as the first step of the pipeline.

After calculating the logarithm of the Fourier power spectrum on each frame, a filter-bank is applied to obtain energy coefficients of the frequency bands. In the MFCC case, the bands are spaced according to the Mel scale to better mimic the frequency resolution of the human ear. Cepstrum coefficients are then obtained by applying the discrete

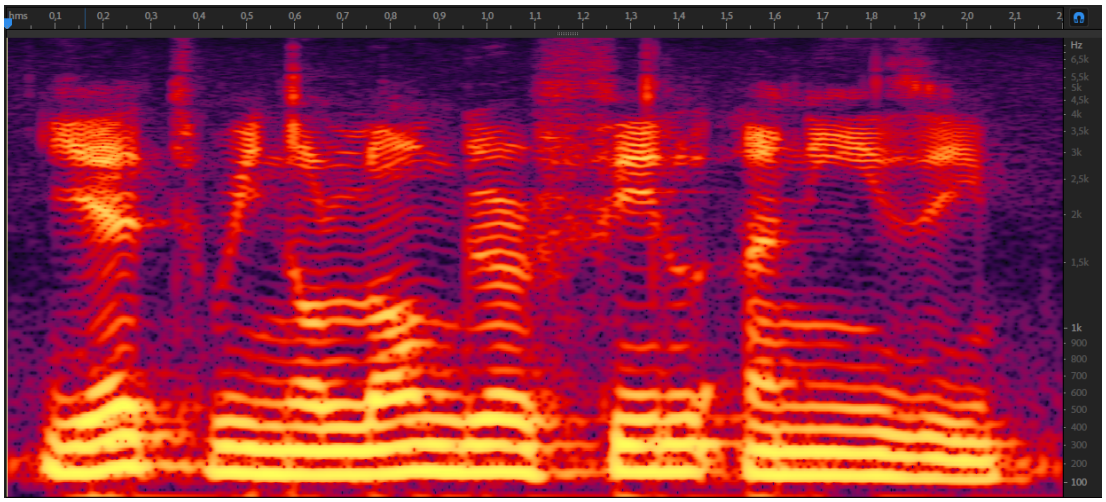


Fig. II.3 *Spectrogram of 2 seconds of speech, sampled at 16 kHz. Brighter colour indicates higher energy.*

cosine transform (DCT) to the filter-bank energies [16], as illustrated in Fig. II.2. The DCT step has the important property of creating highly decorrelated coefficients. This i) makes easier modelling their behaviour statistically and ii) results in a cepstrum with regions representing the energy, the filter and the excitation (pitch) of the vocal tract. Usually only the dominant, filter-related coefficients are kept. The energy coefficient (C0) is usually discarded as features should be energy-invariant. Pitch information is discarded as well, often deemed more harmful than useful as plenty of speakers can share the same pitch. Moreover, the fundamental frequency (F0) of male speakers is often below the lower limit of the telephony spectrum.

Each of the above steps comes with a reduction in dimensionality (and loss of information), and the whole process goes from a mono-dimensional highly correlated time signal to a 2-dimensional compact representation whose elements are assumed to be independent across both dimensions. MFCCs are often enhanced by appending first and second order time derivatives for each coefficient, known as *deltas* and *double deltas* [11]. These add some information about the dynamics of speech and can be calculated as simple differences with the previous frame, or weighted sums of several previous and future frames.

II.2.2 Longer-term features

Although short-term features proved to be the most popular feature type, it is widely known that useful speaker information resides in phenomena observable outside the 10-30 millisecond interval. Some like *voice source* features, aim to describe parameters of human vocal tract behaviour like the glottal pulse shape and F0 (the rate of the vocal fold vibration) of voiced sounds [11].

Other longer-term features observable across 100-500 millisecond intervals are usually derived from the spectrogram of the signal, hence the name *spectro-temporal* features. The spectrogram representation is obtained as a simple concatenation of frame spectra

and forms a 2-dimensional time-frequency representation, allowing extraction of speaker-specific informations like the trajectory of formant frequencies and *coarticulation* [1]. This image-like (see Fig. II.3) but still highly-dimensional representation is what made the spectrogram itself the low-level signal fed to *deep learning* systems which had their research roots in image recognition. The topics of *neural networks* and *deep learning* will be discussed in a dedicated literature review in chapter V.

Other non-segmental features like *prosodic* and *idiolectal* are statistic in nature and aim to model both instantaneous and long-term information. F0 itself is a prosodic feature when its variation is modelled over words or entire sentences, others like syllable stress and speaking rate are useful descriptors of the speaker talking style. Occurrences of certain words or patterns are called idiolectal features and characterise the speaker at a higher, conversational level [2]. The higher the level of features, the longer the window across which such features can be observed, the larger the amount of speaker data needed to provide solid statistics.

II.3 Back end: models and classifiers

The term *modelling* has been used very loosely throughout this chapter to describe the search for cues in the feature space, aiming to describe a speaker as "unique" as well as different from the impostors. These two adjectives might seem to carry the same concept but they are at the core of the two families of machine learning modelling techniques: *generative* and *discriminative* models [11].

Generative modelling implies one model for the target class and one for the impostors (usually referred as background). Each model is built using only data from their respective classes. At inference time, the trial utterance is considered to belong to the class it is estimated to be closer to. Generative models are often paired with *supervised* training because, understandably, data must be labelled in order to use it to train the appropriate model.

Discriminative modelling involves only one model. Its job is to separate the two classes by learning what makes them different, akin to tracing a boundary between features corresponding to each class. At testing, the model "places" the trial utterance on the estimated correct side. Discriminative models can also be trained in an *unsupervised* fashion: even with no ground truth available, the model can cluster data into classes by exploiting the underlying differences.

II.3.1 Gaussian Mixture Models

Gaussian mixture models (GMMs) were used for the first time for speaker recognition in [17] in 1995 and replaced Vector Quantization techniques as the state-of-the-art method for speaker modelling. They since became the basis of several GMM-derived approaches discuss later in this Section. A GMM models the speaker with a mixture of Gaussian probability density functions (PDFs) calculated on the coefficients of the features. The GMM approach belongs to the generative modeling category and uses the expectation-maximization (EM) algorithm [18] to iteratively move the initially random parameters

of the distributions closer to those of observed data. The probability of an observation vector x_t belonging to model λ is calculated as:

$$p(x_t|\lambda) = \sum_{g=1}^{G} \pi_g \mathcal{N}(x_t; \mu_g, \Sigma_g) \tag{II.1}$$

where the parameters are the mean vectors (μ) , covariance matrices (Σ) and weights (π) of each of the G multivariate Gaussian components.

A probability score is obtained by calculating the *likelihood* of the test features with respect to the target Gaussian mixture, which is a proximity measure of the test utterance to the speaker model. Since feature frames are assumed to be statistically independent, a test utterance can be treated as a sequence of independent observations $X = \{x_t|t \in 1...T\}$. Their likelihood is calculated as:

$$P(X|\lambda) = \prod_{t=1}^{T} P(x_t|\lambda) \tag{II.2}$$

When applied to speaker authentication, the score is obtained by the ratio of the aforementioned model log-likelihood with the log-likelihood of the test features with a *Universal Background Model* (UBM), obtaining the *log-likelihood ratio* (LLR). The UBM is also a GMM, trained with a higher amount of data of a large number of speakers and represents a generic model of any speaker except the target one [19]. By defining:

- X as the observation (the test trial features)
- λ as the target GMM
- β as the UBM
- H_0 as the positive hypothesis (X was spoken by the target speaker)
- H_1 as the alternative hypothesis (X was not spoken by the target speaker)

The log-likelihood ratio is obtained as:

$$\text{LLR} = \log \frac{p(X|H_0)}{p(X|H_1)} = \log \frac{p(X|\lambda)}{p(X|\beta)} \tag{II.3}$$

where $p(X|\lambda)$ and $p(X|\beta)$ are calculated as in Eq. II.2. Neither GMM training nor test utterances need to be constrained to a specific duration. The order of the features does not matter. In fact, GMMs are not capable of modelling the time dimension. The latter is a not a great issue for text-independent tasks, but can be a disadvantage when using plain GMM models for text-dependent speaker authentication.

II.3.2 GMM-UBM

In traditional GMM-based approaches both the speaker GMM and the UBM are trained with the EM algorithm. In 2000, the advancement in performance achieved by the so-called GMM-UBM method was obtained by using the UBM as a starting point to derive the GMM, adapting the parameters of the former [20]. This way, the UBM acts more as a well-trained *speech* or *voice* model, representing characteristics common to any human speaker, which are then adapted to better resemble those of a target speaker. This is achieved through *maximum a posteriori* (MAP) adaptation [21].

A fundamental parameter of the MAP algorithm which governs the degree of adaptation is the so-called relevance factor, τ. Together with a probabilistic count of new data n_i for each Gaussian component i, it is used to determine an adaptation coefficient given by:

$$\alpha_i^\rho = \frac{n_i}{n_i + \tau^\rho} \tag{II.4}$$

where $\rho \in \{\omega, \mu, \sigma\}$ indicates the relevance factor for the weight, mean or variance parameters of the GMM. The adaptation coefficients are then used to obtain the new weight, mean and variance estimates according to:

$$\hat{\omega_i} = [\alpha_i^\omega n_i/T + (1 - \alpha_i^\omega)\omega_i]\gamma \tag{II.5}$$

$$\hat{\mu_i} = \alpha_i^\mu E_i(x) + (1 - \alpha_i^\mu)\mu_i \tag{II.6}$$

$$\hat{\sigma_i^2} = \alpha_i^\sigma E_i(x^2) + (1 - \alpha_i^\sigma)(\sigma_i^2 + \mu_i^2) - \mu_i^2 \tag{II.7}$$

where each equation gives a new estimate from a weighted combination of the respective training data posterior statistics with weight α and prior data with weight $(1-\alpha)$. T is a normalization factor for duration effects; γ is a scale factor which ensures the unity sum of weights. $E_i(x)$ and $E_i(x^2)$ are the first and second moments of posterior data whereas μ_i and σ_i^2 are the mean and variance of prior data, respectively.

Adapted GMMs were found to be robust, with error rates reduced to 33% of those using EM-trained GMMs [19], making the approach state of the art of the early 2000s. Later systems (see Section II.3.4) used the GMM-UBM approach as a basis and/or as a reference baseline [1, 11], cementing its legacy in the ASV community.

II.3.3 Hidden Markov Models

Hidden Markov models (HMMs) are well suited for temporal pattern recognition. They have been used to model phoneme transition probabilities in speech recognition tasks [22], and are also useful in ASV, especially in its text-dependent variant where the sequence of observations is more important.

HMMs can enhance GMMs by adding sensitivity to the time domain, they can be adapted from UBMs or speaker GMMs without necessarily going to the phoneme level or adding speech recognition capabilities to the system, as in [3]. HMMs are of key importance for text-dependent systems (see Section II.3.5) and for the contribution described in chapter IV. The feature frames which represent the observations $X = x_t, ..., x_T$ are segmented in N states along the time axis. The states can be modelled by N different

GMMs with associated transition probabilities to form a temporal model of the speaker and the given sentence. This way, the order of the features actually matters (*i.e* it would not be possible to obtain high likelihoods at test by uttering the same sentence with words in a different order, an aspect to which plain GMMs are just insensitive to). An N-state HMM model is defined by:

- A state-transition probability matrix $A = \{a_{ij}\}$
- An observation PDF matrix $B = \{b_i(x_t)\}$
- An initial-state probability vector $\eta = \{\eta_i\}$

When used to model a known sentence, η is initialised to 1 for the first state and zero for the rest, and A is limited to only allow transitions to the current or next state. Initially, training examples are divided into N equally long chunks. The data in each chunk is used to model the corresponding HMM state in the form of a GMM, with its μ, Σ and π parameters (see II.3.1). Each HMM state, not being constrained to defined speech units like phonemes, can encompass syllables or even words depending on their number and the sentence duration. The initial equally-divided frame assignment is thus suboptimal because of inter-utterance variations of the training examples. To assign feature frames so that each state is trained only with frames pertaining to the same speech units across all training utterances, unsupervised Viterbi realignment [23] can be iteratively applied. Define P as the probability of observing X given state sequence s over all possible S state sequences:

$$\sum_{S} P(X, s|\gamma) = \sum_{S} \prod_{t=1}^{T} a_{s_{t-1},s_t} b_{s_t}(x_t) \tag{II.8}$$

The *Viterbi path* is the most likely sequence of hidden states $s' = \{s'_1, ..., s'_T\}$ for observation X:

$$s' = \text{argmax}_s P(X, s|\gamma) \tag{II.9}$$

The realignment algorithm consists of two iterative steps:

1. For each training example, the Viterbi path for the HMM is calculated and frames are re-assigned to the N states.
2. The μ, Σ and π parameters of each HMM state are then updated according to the new state occupancies.

The algorithm can be set to stop after a predefined number of iterations or when the frame assignments are unchanged with respect to the previous iteration. At test time, the likelihood of the trial observation X given the trained HMM model γ is equal to the probability of the Viterbi path s', calculated as $P(X, s'|\gamma)$ according to Eq. II.8.

In speaker verification, HMMs have been used to model entire sentences [24] as well as single words [25]. HMM sensitivity to the time domain has been proven to be an

advantage, especially when there is text mismatch between the model and the test trial: the study in [26] shows a relative 14% decrease in error rates when using HMMs over GMM-UBM to model single words.

II.3.4 Towards i-vectors

By concatenating the mean vectors of a MAP-adapted GMM speaker model (see Section II.3.2), a fixed-dimensional representation of a variable-duration utterance is obtained in the form of a *GMM supervector* [27]. *Support Vector Machines* (SVMs), which were already successful as supervised binary classifiers in machine learning [28] found their application in speaker verification tasks [29] by letting them operate in the supervector domain. The discriminative nature of SVMs allowed them to be trained with GMM supervectors derived from (labeled) target and impostors utterances, using them as features.

Several other techniques were applied to work in the supervector domain, most of them belonging to the family of factor analysis approaches (FA). FA aims at giving insights to the speaker-and-channel variabilities in the highly dimensional supervector space in a more compact way with fewer hidden variables which can be estimated. All of the FA-derived techniques describe GMM supervectors as statistical models consisting of linear combinations of (i) speaker- and (ii) channel-dependent components, (iii) a channel-and-speaker independent component and (iv) a residual. In fact, even the previously discussed MAP adaptation (see II.3.2) can be interpreted as linear statistical modelling [11]. Most notable FA approaches include *joint FA* and *i-vectors*.

Joint factor analysis (JFA) managed to model both speaker and channel variability in one model through MAP adaptation, assuming these variabilities lie in distinct lower-dimensional subspaces and takes into account all four previously mentioned linear components [30]. When JFA is used in conjunction with a SVM classifier, the estimated speaker and channel factors are used as features and referred to as i-vectors [31]. I-vectors are therefore related to both speaker and channel variability without distinction like the initial GMM supervector. With their reduced dimensionality they allow for compensations techniques to be easily applied.

Probabilistic Linear Discriminant Analysis (PLDA) [32] is to i-vectors what JFA is to supervectors. At the risk of oversimplifying, they both aim to model speaker- and channel-dependent variations in a lower-dimensional subspace. The reduced numbers of PLDA parameters compared to JFA, less demanding computational efforts and the possibility to train the i-vector extractor on unlabeled data sets made the i-vector/PLDA method appealing and the approach raised to state-of-the-art, reporting error rates as low as 1.27% for the core condition of the NIST Speaker Recognition Evaluation of 2010 (see Section II.5.1).

While developed for text-independent tasks, i-vector/PLDA systems have been shown to benefit from lexical information during training [33]. Their inability to model the temporal structure of the utterances, however, makes them ill-suited for text-dependent tasks, especially when dealing with short utterances [34]. The work in [35] reports comparative experiments between the HMM-based system HiLAM (explained in the next section) and an i-vector system: on the matched-text condition of RSR2015 corpus (often deemed the most difficult, see II.5.2.b) i-vector were always outperformed by HiLAM, with the closest

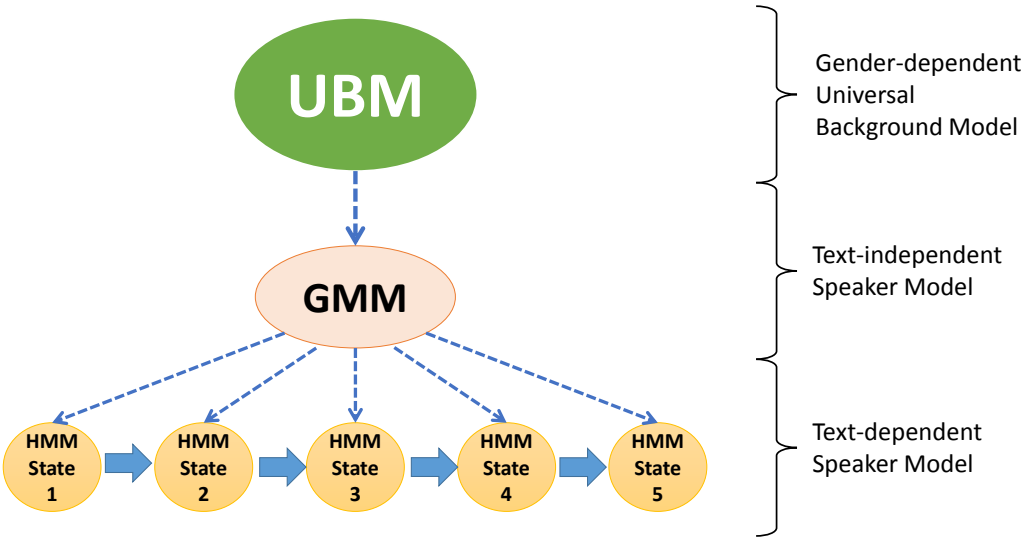


Fig. II.4 *The original HiLAM system architecture reproduced from [37].*

gap being a 30% relative error difference on male speakers of part II of the database.

II.3.5 The HiLAM system

HiLAM (Hierarchical multi-Layer Acoustic Model) [35], is a text-dependent ASV system which is here given particular attention since it forms the basis for contributions explained in chapters III and IV.

After its introduction in 2014, the system quickly became the state of the art for short-utterance text-dependent speaker authentication (similar systems were previously proposed by the same author [24,36] with the first mention of "HiLAM" actually appearing in [37]. None of the block of its pipeline made use of current cutting-edge or experimental technologies, it was indeed the judicious combination of traditional components like MFCCs GMM-UBM and HMMs that, when used on a sub-field of ASV (namely short-utterance text-dependent speaker authentication) made the approach outperform other current approaches (see section II.3.4).

The HiLAM system is a flexible, efficient and competitive approach to text-dependent automatic speaker verification which made it a perfect starting point for the research on embedded applications which is reported in this thesis. The architecture is illustrated in Fig. II.4 and is composed of three distinct layers. They represent (i) a gender-dependent UBM, (ii) a text-independent GMM speaker model and (iii) a text-dependent HMM speaker model.

The UBM is trained according to a conventional maximum likelihood / expectation maximization algorithm [38]. The second-layer text-independent speaker model is derived from the UBM via MAP adaptation (see II.3.2). Different third-layer text-dependent speaker models are then learned for each sentence or pass-phrase. These take the form of 5-state, left-to-right HMMs. Each state of the HMM is also MAP-adapted from the second

layer text-independent GMM of the corresponding speaker and then learned with several iterations of Viterbi realignment and retraining (see Section II.3.3). Each HMM therefore captures both speaker characteristics in addition to the time-sequence information which characterizes the sentence or pass-phrase.

Given the 3-layer structure, to obtain a single text-dependent model MAP adaptation needs to be performed twice: once for the adaptation of the UBM to the GMM and a second time for the adaptation of the GMM to the HMM. Two distinct relevance factors are therefore applied during the process, τ_1 and τ_2. The first relevance factor, τ_1, acts to balance the contribution of the UBM and speaker-specific adaptation data to the parameters of the new speaker model. The second, τ_2, controls adaptation between the text-independent and text-dependent speaker models. Of course, since the same text-independent model can be used to adapt several sentences, the first adaptation step is performed just once per speaker while the second is performed for each speaker and pass-phrase.

II.4 Performance Metrics

This section describes the metrics that are used (directly or as a basis) for performance assessment in the remainder of this thesis. Fig. II.6 illustrates an example of Gaussian distributions of impostor and target scores. To calculate any performance a decision threshold must be set, which represents the minimum score a trial must obtain to be considered a target.

In practice, some overlap between the distributions will always be present, which implies that any threshold would cause at least one of two types of errors: *false positives* and *false negatives*. The former is generated from impostor trials with scores above the threshold and the latter, conversely, from target trials with scores below the threshold. In real case scenarios, the threshold is set a priori and should be fixed. Threshold tuning is a matter of compromise between more false positives or negatives, the value is set in function of their estimated probabilities or based on the type of application for the system. When a system is in its development stage, performance metrics are reported as a function of the decision threshold, or the threshold can be set a posteriori.

II.4.1 Receiver Operating Characteristic (ROC)

The ROC is explained here instead of the more widely used (in ASV) *detection error tradeoff* (DET) curve, because it is of fundamental importance for contributions in chapters VI and VII.

By defining the *true positive rate* (TPR) as:

$$TPR = \frac{\#\{\text{true positives}\}}{\#\{\text{actually positives}\}} = \frac{\#\{\text{true positives}\}}{\#\{\text{true positives}\} + \#\{\text{false negatives}\}} \tag{II.10}$$

and the *false positive rate* (FPR) as:

$$FPR = \frac{\#\{\text{false positives}\}}{\#\{\text{actually negatives}\}} = \frac{\#\{\text{false positives}\}}{\#\{\text{false positives}\} + \#\{\text{true negatives}\}} \tag{II.11}$$

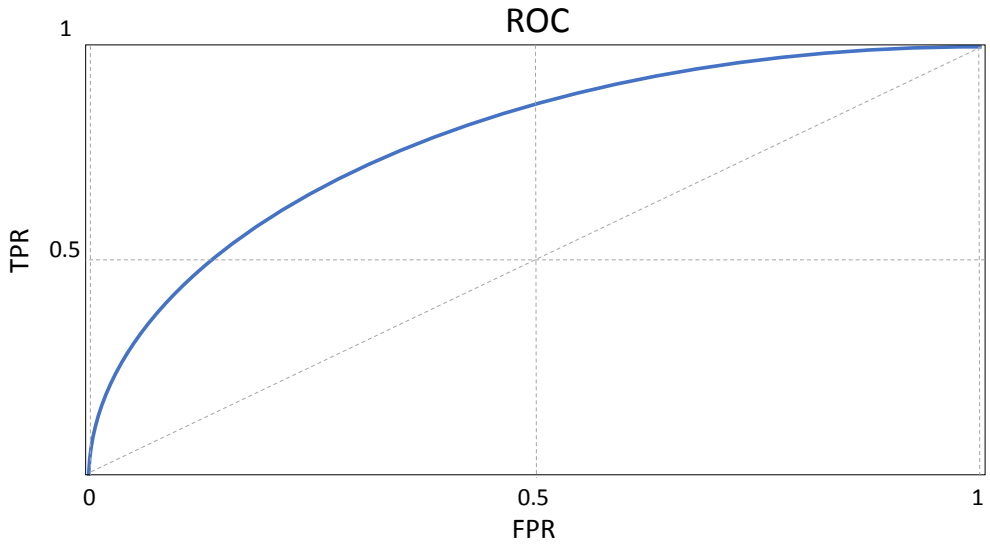


Fig. II.5 *An example of a ROC curve: the profile can be interpreted as the loss in TPR correspondent to a lower FPR and vice-versa. Better performance corresponds to curves with faster acceleration, pushed to the upper-left corner of the graph.*

The ROC curve (see Fig. II.5) is depicted by plotting the FPR versus the TPR variations as a function of the decision threshold of a binary classifier. It conveys the same type of information as DET graph, albeit with linearly-scaled axes and the use of the *false negative rate* in lieu of the TPR.

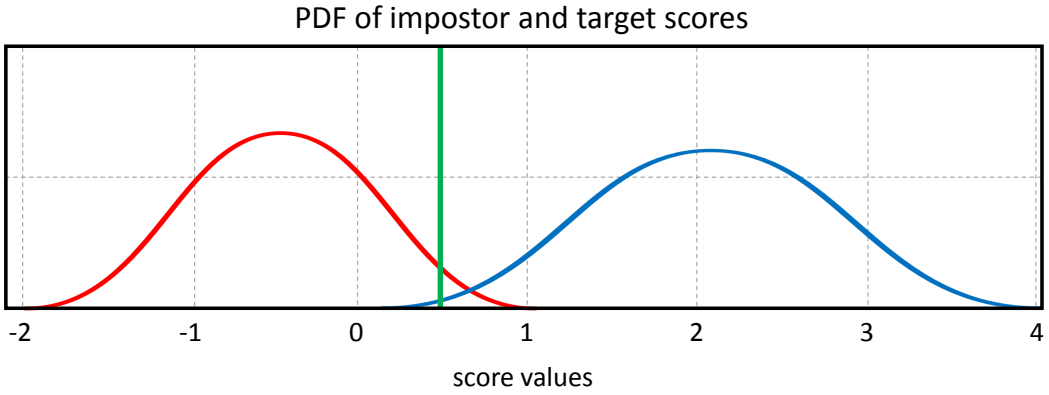


Fig. II.6 *Example of impostor (left) and target (right) scores probability density functions (PDF). The decision threshold is depicted as the green vertical line.*

II.4.2 Equal Error Rate (EER)

The equal error rate is the point on ROC or DET curves for which both error rates (false positive and false negative) are equal, identified by one threshold value. As a consequence, the EER threshold falls roughly in the middle of the overlap area of score distributions, as exemplified by the green line in Fig. II.6.

While in real-case scenarios the two error types are rarely considered equally severe or equally probable (and therefore systems are tuned to achieve often distinct error rates) the EER remains a compact and balanced descriptor of a system overall performance and it is often used as an optimization objective [39]. In all this thesis contributions performance is reported in the form of EER.

II.4.3 Score normalisation

Studies like [40] demonstrated that different speakers exhibited different target and impostor score distributions, which penalized performance assessed using a single decision threshold over all trials. Score normalisation techniques were developed as a remedy, the most notable being *Z-norm* (Zero normalisation) and *T-norm* (Test normalisation) [2]. Z-norm compensates inter-speaker variation by testing a target model against an external cohort of impostor trials and using its mean and variance to normalise the actual scores, approximating a common impostor distribution to all speakers. T-norm compensates inter-session variations by testing the trial utterance against a fixed set of impostor models, and using the trial-dependent statistics to normalise the score in an online fashion, achieving better separation between the speakers target and impostor distributions. The two normalisation techniques can be used jointly, the practice is known as *ZT-norm*.

II.5 Challenges and Databases

Performance measures and error rates would not be meaningful if different systems are not compared using the same dataset and the same experimental protocols. This is why since the birth of ASV great effort has been put in creating standard databases and protocols.

II.5.1 NIST Speaker Recognition Evaluations

Text-independent ASV research has largely been driven by the Speaker Recognition Evaluations (SREs) administered by the US National Institute of Standards and Technology (NIST)[1]. Since 1996, there have been 16 evaluations (the 17th is planned for 2019 at the time of writing). With each SRE, an evaluation plan is presented, which describes the rules and challenges for the current evaluation, and is often accompanied by an appropriate corpus. An overview of more than two decades of NIST evaluations can give an idea of the challenges that ASV technologies faced over the years, which approaches went from experimental to state-of-the-art to obsolete, what rules and conditions were added in newer evaluations to add realistic difficulties to the tasks or embrace specific scenarios [1]. This section will focus on the most notable challenges introduced in the ASV community by the NIST evaluations.

[1] https://www.nist.gov/itl/iad/mig/speaker-recognition

II.5.1.a The early years

NIST SREs up to the early 2000s involved speaker authentication over telephone conversations. These were extracted from the *Switchboard* corpus and had an average length of 5 minutes. The number of conversations and speakers were in the order of thousands and hundreds, respectively. Training data usually consisted in one or two sessions. The state-of-the-art approach of the time were GMM and GMM-UBM based systems which worked on short-time cepstral features like MFCCs. The main weaknesses for those systems were the high sensitivity to channel variations (mainly due to different telephone microphones) and the inability to benefit from test trials with more than one minute duration (conversely, strong degradation occurred with very short durations) [41]. It was also in this period that score normalisation techniques (see Section II.4.3) were introduced.

II.5.1.b Broader scope and higher dimensionality

The mid-2000s saw expansions in every aspect of previous evaluations: the number of speakers, sessions, languages, microphone types, conditions and tasks were all increased. Most notably, up to 16 sessions were available to train each speaker, totalling 40 minutes of data, with training conditions starting from 10 seconds to the full session set. The *Mixer corpora* consisted of bilingual speakers fluent in a second language (the first always being English) to allow for language matched and mismatched trials.

The increase in amount of data available came with systems that could benefit from it. Peculiar to this period is the attention given to longer-term, non-cepstral features, namely *phone-sequences*, *prosodic*, *lexical* and *conversational* features, which all exploit traits of speech that cannot be captured by frames that are a few of milliseconds in duration. Describing the systems that employed these features is outside of the scope of this thesis, but it is worth noting that they achieved performances equal to the reference GMM-UBM, when classifier fusion was applied. Standard short-term cepstral features found new application in high-dimensional spaces with discriminative GMM-SVM approaches (see Section II.3.4), yielding significant improvements in handling channel variability.

II.5.1.c Bi-annual big data challenges

Each of the four NIST evaluations from 2006 to 2012 introduced new challenges, rules and tasks, as well as copious amounts of new data to exploit. Speakers were now in the order of thousands, generating hundreds of thousands of trials. Systems were trained with several hours of speech. SRE08 saw the introduction of the *interview* scenario, bringing high-quality and overall different kind of conversational speech to the table. In 2010 new telephone data was recorded and divided in high and low vocal effort; the same year also saw the introduction of the new detection cost function. 2012 was the year of the perhaps biggest departure: the amount of speech to train a given model was no longer limited by the protocols: any combination of previous evaluation files could now be used. New challenges were added in the form of environmental and additive noise, and the unavailability of speech-recognition word transcripts.

Focus was shifted back to cepstral-only systems, with JFA-based systems obtaining excellent results in several SREs only to be overthrown by the i-vector/PLDA approach in 2012 (see Section II.3.4.)

Table II.1: *Statistics for the NIST SRE16 development labelled data*

# Speakers	20
# Models	80
# Calls	200
# Target trials	4828
# Impostor trials	19312
Languages	Cebuano Mandarin

II.5.1.d SRE16

The recent NIST SRE16[2] is given special attention here since it is used for experiments reported in Chapter VI. The main focus of this evaluation is telephone speech, with more duration variability compared to previous SREs. Data is collected from the *Call My Net Speech Collection* and is comprised of speech samples in the Tagalog, Cantonese, Cebuano and Mandarin languages.

The required **fixed** training condition poses limits to the data that can be used to train the system which —apart from the corpus pertaining to the evaluation— can only consist of previous SRE evaluations, Switchboard (only files with transcriptions) and Fisher corpora datasets. This, along with the aforementioned language choices, makes background-garbage modelling particularly challenging, as most of the speech from these corpora is in English. The enrolment data for a given speaker model can be either one or three segments of telephone conversations (same handset), each segment containing approximately 60 seconds of speech. Test segments durations range from 0.5 to 60 seconds, recorded from several handsets including the enrolment one. Durations measurements refer to detected speech counting both sides, not the actual recording length.

Data for the *development set* is collected in a way that aims at mirroring the *evaluation set* conditions. The statistics for the labelled part are described in Table II.1.

Each of the 20 speakers is featured in 10 calls with an average recording duration of approximately 8 minutes. As can be seen in Table II.1, speakers enrol multiple models following the 1- and 3-segments conditions described above. The testing protocol involves segments by the same 20 speakers (non overlapping with enrolment data), with an average of 75 and 225 target and impostor trials per model, respectively.

While most SREs have included conditions for training and testing durations as short as 10 seconds, optimisation efforts were always towards better results in standard conditions. This often resulted in state-of-the-art systems having sharp degradations in performance on short duration tasks [42, 43]. In [44] experiments on NIST 2008 and 2010 databases with truncated data showed that, for durations below 10 seconds, state-of-the-art i-vector systems had no significant advantage over GMM-UBM systems. Notably, the latter even performed better with both training and testing truncated to 2 seconds, a desirable condition in IoT scenarios (see Section II.3.5 and Chapter III).

[2] https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016

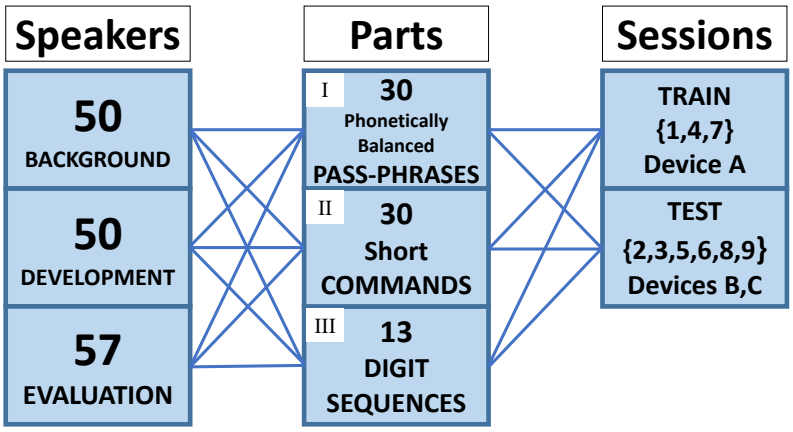


Fig. II.7 *RSR2015 database partition for male speakers. The partition is identical for female speakers but with only 43 speakers in the evaluation set.*

II.5.2 RSR2015 corpus

The RSR2015 corpus [37] is a short-utterance, text-dependent database. Its male gender partition and related protocols described below were used extensively for the work reported in Chapters III and IV. RSR2015 was released almost in tandem with the HiLAM system presented in Section II.3.5; in fact, most of the experimental work involving HiLAM was performed on this corpus [35, 45] which is nonetheless distributed with protocols suited to the assessment of HiLAM-based text-dependent speaker verification systems. The RSR2015 database is one of the most versatile and comprehensive databases for such research. The particular speaker/part/session all-combinations structure illustrated in Fig. II.7 is what made RSR well suited to the work reported in Chapters III and IV. The more recent RedDots [46] corpus does not reflect this structure.

RSR2015 contains speech data collected from both male and female speakers and is partitioned into 3 evenly-sized subsets whose usual purpose is for background modelling, experimental development and evaluation. Each subset is comprised of 3 parts: phonetically-balanced sentences (part I), short commands (part II) and random digits (part III). Each part contains data collected in one of nine sessions. Three of these sessions are reserved for training while the remaining six are set aside for testing. The three training sessions are recorded using the same smart device whereas the six testing sessions are recorded using two different smart devices (i.e. the user kept the same mobile phone or tablet for the training sessions while two other were used for the testing sessions, but the devices themselves differ between users).

II.5.2.a Training

When used in conjunction with the HiLAM system, the background speakers are used solely to build the UBM. Not all the data is used, though: in order to have no speaker nor text content overlap, the UBM should be built avoiding data from the part which would be used to train the other layers. For example, consider the end goal is to test speaker recognition accuracy on phonetically balanced pass-phrases (part I). First, the

Table II.2: *The four different trial types used to assess the performance of a text-dependent speaker verification system. They involve different combinations of matching speakers and text. A trial should be accepted only when both match.*

Trial Type	Speaker Match	Text Match
Target-Correct (TC)	Yes	Yes
Target-Wrong (TW)	Yes	No
Impostor-Correct (IC)	No	Yes
Impostor-Wrong (IW)	No	No

Table II.3: *Number of trials for Part I of the RSR2015 database for each of the four trial types illustrated in Table II.2 and for development (Dev) and evaluation (Eval) subsets.*

Speaker-Text	Dev	Eval
Target-Correct (TC)	8,931	10,244
Target-Wrong (TW)	259,001	297,076
Impostor-Correct (IC)	437,631	573,664
Impostor-Wrong (IW)	6,342,019	8,318,132

UBM should be built using background speakers only from part II [37]. Then, the training sessions from part I would be used to adapt the middle-layer GMMs of non-background speakers. The text-and-speaker dependent data used to adapt each HMM model is a subset of the training sessions used for the upper layer.

II.5.2.b Testing

Being a text-dependent corpus, there are actually four possible trial types, these are illustrated in Table II.2. To be a target trial, **both** the speaker and the text content must match the ones of the claimed model, dubbed a *target-correct* trial. The remaining three trial types (*target-wrong, impostor-correct, impostor-wrong*) are considered impostor trials. Performance is usually assessed separately for the three impostor conditions, combining target-correct (TC) trials with trials of **one** mismatching combination: target-wrong (TW), impostor-correct (IC) or impostor-wrong (IW). The number of trials for each type in the standard RSR2015 protocols is illustrated in Table II.3 for development and evaluation sets. The number of trials for each testing condition is TC+TW, TC+IC and TC+IW respectively.

II.6 Summary

This chapter presented an overview of Automatic Speaker Verification as a research field, describing its variants and its traditional front-end back-end pipeline. Historically important ASV approaches were reviewed, most of which are still relevant today and are at

the basis of current systems. Neural-network based approaches deserve their own literature and will be explored in Chapter V. Evaluations (and their evolution) and databases for both text-independent and text-dependent tasks were analysed. These ASV topics formed the starting point for the author's research activity and acts as a basic knowledge on which the contributions in the following chapters III and IV are built upon.

Chapter III
Simplified HiLAM

This chapter is concerned with increasing the *usability* of the HiLAM system, this is achieved by studying the effect of progressively pruned training data on final performance. The goal is to obtain a system that requires less data (and therefore less time) to enrol a speaker, to the point where it is acceptable for an end-user in a real case scenario.

During the last decade, text-dependent speaker authentication became known to the general public as it slowly entered everyday life [47]. Voice commands in general are perceived as quick, non-intrusive and efficient. Online banking services started to employ fixed-text speaker authentication and several companies are selling voice assistants. Though most of the latter only perform speech recognition, embedding speaker authentication seamlessly in the sentence used to "wake up" the device (*i.e.* "Hello Google" or "Alexa?") is a highly desirable feature.

As introduced in previous chapter, the NIST SREs have been strong driving forces of ASV research during the last 20 years. Their main focus being security and surveillance, NIST challenges often involve text-independent, relatively long telephone conversations. Text-dependent speaker authentication scenarios usually calls for very short test trials. This is possible, due to the fact that locking the system to a fixed sentence greatly decreases the amount of variability in training examples and allows the system to know "what to expect" at test time, in terms of phonetic content. Nevertheless, significant reductions in speaker-specific data (both at enrolment and at test time) pose several challenges to state-of-the-art ASV technologies [42–44]. State-of-the-art i-Vector and probabilistic linear discriminant analysis (PLDA) techniques are difficult to apply in text-dependent tasks [35, 48, 49] unless training data is plentiful [50] and unless impostor trials involve matching text [51]. Studies reported in [52–55] demonstrate that joint factor analysis (JFA) systems can work well with little enrolment data, however, even under those conditions, both JFA and PLDA still rely on prior knowledge of the text content.

The HiLAM system described in II.3.5 represents a good candidate for such IoT-related scenarios, as it allows the user to train any short command or pass-phrase (within seconds length), all of which will embed speaker authentication as the final model is content-and-speaker specific. HiLAM involves two speaker training stages: one to train the text-independent model, and a second to train the text-dependent model. The only problem here is that, in its presented form, the standard pipeline requires roughly 5 minutes of users' speech to train the text-independent model, prior to modelling a specific pass-phrase. Pass-phrase can be modelled with only three utterances. Five minutes is a

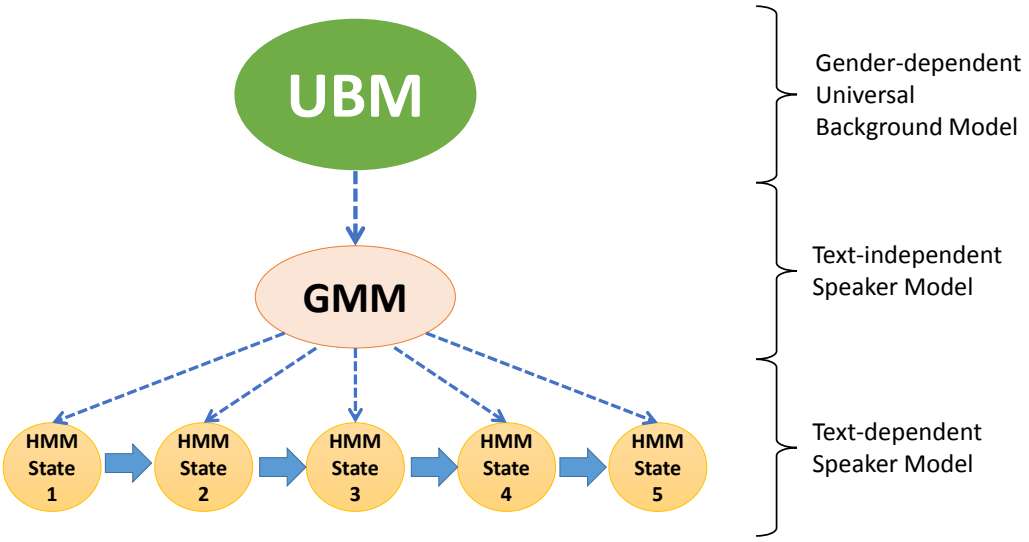


Fig. III.1 *The original HiLAM system architecture reproduced from [37]. This graph is the same used in previous chapter, Fig. II.4; it is again reproduced here to ease the reading of this chapter.*

rather long time to ask as a preparatory step, even if it's only required once per speaker. It would be perceived as very inconvenient and most people will probably not bother to do the effort. Moreover, the text-independent models are not even needed at test time.

How much can those 5 minutes be reduced? Is there a good compromise between usability and performance? The work reported in this chapter demonstrates that, by sacrificing a little in performance, great improvements can be obtained in terms of in usability and efficiency. The modifications introduced to HiLAM eliminate the need for the intermediate text-independent speaker model.

Albeit a modest modification to the original work, the simplified 2-layer approach leads to significant reductions in the demand for enrolment data making the system more efficient, supporting speaker authentication for smart device and Internet of Things applications. The contributions derived from this work were originally published in [4].

III.1 HiLAM baseline implementation

This section describes our implementation of the HiLAM system that forms the baseline for the work reported here. The system architecture is illustrated in Fig. III.1. Also presented are results for our specific implementation assessed using the RSR2015 database.

III.1.1 Preprocessing and feature extraction

Silence removal is first applied to raw speech signals sampled at 16 kHz. This is performed according to ITU-T recommendation P.56[1] which specifies an active speech level of 15.9

[1] http://www.itu.int/rec/T-REC-P.56-201112-I/en

dB. In our implementation this results in the removal of approximately 36% of the original data. The remaining 64% is then framed in blocks of 20ms with 10ms overlap. The feature extraction process is standard and results in 19 static Mel frequency cepstral coefficients (MFCC) without energy (C0). These are appended with delta and double-delta coefficients resulting in feature vectors of 57 dimensions.

III.1.2 GMM optimisation

The number of Gaussian components is empirically optimised. The literature shows that higher values (512-2048) are often used for text-independent tasks [20, 56] or with systems based on i-Vector and PLDA techniques [51, 57]. In contrast, lower values (128-256) are typically used in text-dependent tasks and techniques such as HiLAM [24, 37]. We obtained the best performance with 64 Gaussian components.

III.1.3 Relevance factor optimisation

Concerning the relevance factor of MAP adaptation (see Section II.3.2 and Eq. II.4), the best performance is delivered with comparatively higher and lower values of τ_1 and τ_2 respectively. More precisely, τ_1 was set to 19, still inside what is considered to be the "insensitive" interval (8-20) according to the literature [20]; τ_2 was set to 3, as lower values are usually better suited for text-dependent tasks [58]. This means that during both adaptation stages, the middle layer model/data are given less weight. At each MAP adaptation stage, the new weight, mean and variance estimates share the same relevance factor (see Equations II.5, II.6 and II.7).

Scoring can be obtained by calculating the probability of the observation given the HMM model as explained in II.3.3, but the best results were obtained by averaging the log-likelihood ratios (between the claimed text-dependent speaker model and the UBM) across the five states.

III.1.4 Baseline performance

Results for our implementation of the HiLAM baseline are presented in Table III.2 alongside those presented in the original work [59]. Results are presented for male speakers only, for the three different test conditions, namely impostor-correct, target-wrong and impostor-wrong (see Section II.5.2.b).

In the literature the IC condition is considered to be both the most difficult and the most crucial. The latter is quite obvious, as a speaker authentication system should be robust even if the impostor is somehow aware of the expected text content; not without reason this type of trial is sometimes called *sly imposture*. TW trials often prove to be less difficult than IC ones; TW impostures are usually considered the least severe menace since they essentially represent the user not remembering the text content or mismatched commands in an IoT environment at worst (a weakness that could be easily remedied by adding speech recognition capabilities to the system [3], which is outside of our research scope). Lastly, IW trials usually exhibit very low error rates, being different in both aspects to the claim model, and therefore considered to be just *naive impostures*, as they are sometimes referred to.

While results for our system are worse than those in the original work, performance is still respectable on IC and IW conditions with respective EERs of less than 2% and 1%, for both development and evaluation subsets. The TW condition showed abnormally high EER values which do not follow the aforementioned trends. Considerable time has been spent to investigate the reason for this difference, one factor was found to stem from the use of delta features. Since deltas and double-deltas represent 38 of the 57 MFCC coefficients, during the MAP adaptation step from the second to the third layer they are believed to introduce noise which degrades the text-dependent modelling capability of the HMMs. Since TW trials are anyway uttered by the target speaker, this causes more false positives. This is confirmed by the fact that, without deltas, TW EERs lie in between those of IC and IW values as expected, but the results are overall much worse. Despite experiments to tune and optimise the MFCC extraction and normalisation parameters, no compromise was found that allowed good TW results without affecting too much the other conditions. This problem however does not at all influence the final goal of this work, as the final simplified system is inherently immune to delta-caused noise.

III.2 Protocols

Since our target application relates to short-duration pass-phrases, all the experiments in this chapter were performed using part I data of the RSR2015 corpus consisting of phonetically-balanced sentences (see section II.5.2). These are the same 30 Harvard sentences used in the collection of the better-known TIMIT database [35] which were designed to give a broad coverage of phonemes in the English language.

As previously explained in II.5.2.a, data reserved for background modelling is disjoint from training and testing data: when the development and evaluation sets of part I are used for training the models, the background set of part II is used to train the UBM (see Fig. II.7); therefore there is no overlap in terms of speakers or sentences. Second-layer HiLAM models (GMMs) are trained with data from all three training sessions and all 30 sentences, totalling 90 utterances. These models are speaker-specific and text-independent, totalling 50 models for the development set and 57 for the evaluation set. Third-layer HiLAM models (HMMs) are trained with the three training utterances corresponding to each specific sentence, for a total of $30 \times 50 = 1500$ and $30 \times 57 = 1710$ pass-phrase-and-speaker specific models for the development and evaluation set, respectively. Second-layer models are MAP-adapted from the UBM, and, in turn, third-layer models are adapted from the respective speaker model from layer 2. The standard protocols were used to implement, optimise and test the HiLAM baseline described in section III.1.

In order to study the usefulness of the training data reserved for the speaker-dependent middle layer, standard protocols are subsampled reducing the number of pass-phrases. In all of the subsampled-protocol experiments, the data required to build the UBM (first layer) and to adapt the HMM pass-phrase models (third layer) is left unchanged as described in the previous paragraph; subsampling is only applied to the data needed to adapt the middle layer models. Those models are indeed just an intermediate step since, once used to adapt third-layer models, they are never used again anywhere in the whole pipeline.

In terms of quantity, it was decided to experiment with 60 and 30 pass-phrases out of

Table III.1: *Performance for different durations of middle layer training. The last row shows results for the simplified HiLAM system with no middle layer at all. Results shown for the RSR2015 development set and for the IC condition.*

	Number of utterances	**EER**
3-Layer	90	1.63%
	60	1.66% (a) 1.64% (b)
	30	1.62% (a) 1.63% (b)
	3	2.33%
2-Layer	3	1.84%

the original 90. This was done in two ways:

a) **Session subsampling:** Select the first 2 sessions or only the first one out of the 3 reserved for training (same phonetic richness, less session variation)

b) **Pass-phrase subsampling:** Select the first 20 or 10 pass-phrases in each training session (same session variation, less phonetic richness)

Testing protocols used for all experiments are the standard part I testing protocols distributed with the RSR2015 database. Finally, performance is expressed in terms of the equal error rate (EER).

III.3 Simplified HiLAM

Described in this section are experiments which assess the necessity of text-independent enrolment and a number of modifications to the original HiLAM baseline system which enable competitive performance with greatly reduced durations of speaker enrolment data. Among these modifications is the reduction of the 3-layer approach to only two layers and associated re-optimization. The new system learns text-dependent speaker models using only three training utterances.

III.3.1 Middle-layer training reduction

In order to assess the necessity of text-independent enrolment, a first sequence of experiments was conducted where the number of text-independent utterances used for layer-two training was successively subsampled according to the configurations explained in Section III.2. Secondly, the middle layer was trained with the exact same data that would be later used to adapt the third layer (with this configuration the middle layer is text-dependent and the number of middle-layer models is the same as the third layer HMM models); while a somewhat questionable choice, this configuration allows for just three repetitions of the desired pass-phrase at enrolment time while keeping the 3-layer structure and acts as an intermediate step towards the complete removal of the middle layer.

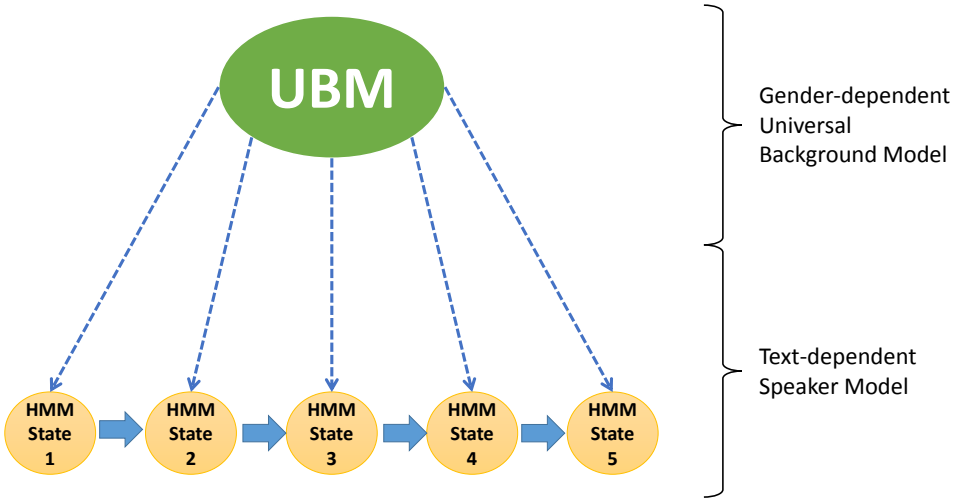


Fig. III.2 *The simplified 2-layer architecture: text-dependent speaker models are adapted directly from the UBM.*

Results are illustrated in Table III.1 for the development set and IC condition only. The first row corresponds to full-protocol training (90 utterances are roughly equivalent to 5 minutes of speech). This result is the same presented in the second row of Table III.2 and acts as the starting point. The number of utterances for the 3-layer system only refers to what is needed to train the middle layer. As the number of training utterance decreases EER values show that performance is fundamentally insensitive to great reductions in the amount of data used for training, whether session subsampling (a) or pass-phrase subsampling (b) is adopted, with one subsampled result yielding an EER even lower than the baseline. Although this last claim is not statistically significant, it is clear at least that the third layer HMM models do not benefit from the increased session variability or phonetic richness of the upper layer. This finding suggests that text-independent enrolment may be unnecessary or perhaps even noisy to some extent, when the recognition task is ultimately text-dependent.

Moreover, the findings relative to the best values for the relevance factor parameters (see Section III.1) indicate that only modest adaptation is applied between layers 1 and 2, whereas more significant adaptation is applied between layers 2 and 3. This calls even more into question the real need for text-independent enrolment or, in other words, the real need for the middle layer.

III.3.2 Middle layer removal

Given the observations reported above, it was decided to assess performance when the middle layer, text-independent enrollment is dispensed with entirely. Speaker enrollment is then performed in text-dependent fashion exclusively as illustrated in Fig. III.2. Each state of the HMM speaker model is now initialized using the UBM instead of the speaker-specific text-independent GMM. Adaptation is otherwise the same as before and performed using the same three utterances of the same sentence. The number of Gaus-

sian components (64) is left unchanged from the 3-layer implementation and the single remaining relevance factor τ (3) is set to the same value of τ_2 (see Section III.1). These parameters were found to be optimal in the case of the simplified system.

Results are illustrated in the last row of Table III.1. Performance degrades from an EER of 1.6% for the baseline 3-layer system to 2.3% when enrolment is performed with only 3 speaker-specific utterances. The exact same data of the latter configuration is used for the 2-layer system, whose performance improves to 1.8% EER. Despite a reduction in enrolment data in the order of 97% with regards to the baseline, the increase in error rate is only 0.2%. Such a compromise between performance and usability would be quite acceptable in many practical scenarios.

III.4 Evaluation Results

Results presented in section III.3 relate to the development set and the IC condition only. Presented in this section is a full performance comparison of the original HiLAM approach in [59] to the simpler 2-layer system presented in this work using the full RSR2015 development and evaluation sets.

Final results for the three impostor conditions now reflect the literature trends described in section III.1 and are illustrated in Table III.2. The first row indicates the specific test condition for development (dev) and evaluation (eval) sets. Results presented in the original work [59] are illustrated in the second row whereas those for the new 2-layer system are presented in the third row. They correspond respectively to the full enrolment condition (90 text-independent utterances for layer 2 and three text-dependent utterances for layer 3) and the reduced enrolment condition (3 text-dependent utterances only). These results confirm the findings presented above, namely that significant improvements to usability can be delivered by reducing the demand for enrolment data with only modest increases in error rates. Both systems achieve better performance for the evaluation set than for the development set. While this finding is counter-intuitive, it is consistent with other results in the literature, e.g. [35,37,59], one possible explanation for which is differences in the distributions of recording devices across the two subsets.

The TW condition results do not exhibit the anomalous behaviour that plagued the baseline system (see Section III.1, last paragraph) and now follow the same trend as the original work. This has probably to do with the fact that, while keeping the same MFCC dimensions, deltas are not adapted from the text-independent model but instead directly from the UBM, which is built from different sentences and more variate speakers.

Compared to the original work, performance for the 2-layer system deteriorates for the development set. In contrast, performance for the evaluation set improves. This result is particularly encouraging. The drop from 1.33% to 1.24% corresponds to a 7% relative reduction in the EER and comes with the same 97% reduction in demand for enrolment data. This is a significant improvement to usability in the case of text-dependent recognition.

Table III.2: *Comparison of results for our implementation of the HiLAM system (3L) with original results reported in [59] and those obtained with the simplified system (2L) reported in this section. Results shown for male speakers in part I of the RSR2015 database. (Results for each condition correspond to their combination with TC trials.)*

System	IC-Dev	TW-Dev	IW-Dev	IC-Eval	TW-Eval	IW-Eval
Larcher 3L [59]	1.43%	1.00%	0.20%	1.33%	0.66%	0.09%
Valenti 3L	1.63%	8.34%	0.78%	1.81%	7.54%	0.83%
Valenti 2L	1.84%	1.09%	0.32%	1.24%	0.52%	0.05%

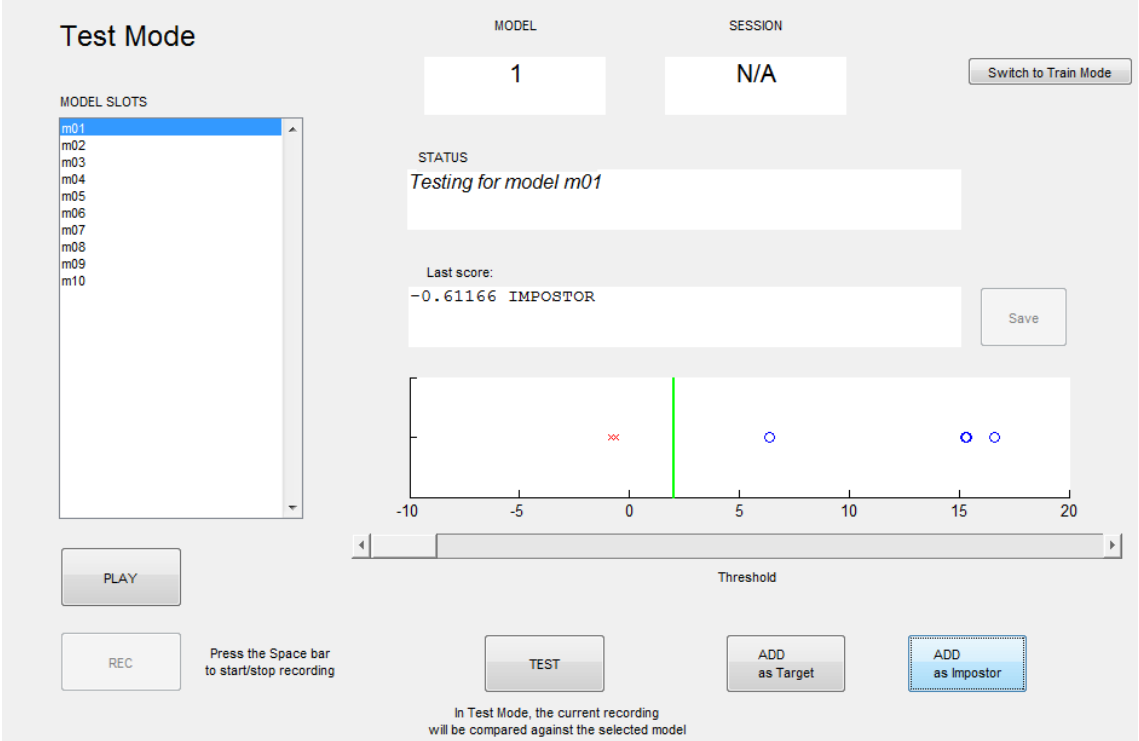


Fig. III.3 *Matlab demo interface in Test Mode: scores are labelled with regards to the threshold, which here is left at its default value of 2. The user can graphically place the scores on the bar by forcing the ground truth symbol of either impostor (red crosses) or target (blue circles)*

III.5 The Matlab demo

Given the increased usability, a demo was developed to show the practical aspects of the above work, in an online fashion. Developed in Matlab and running on standard PCs, the demo was not tailored to any particular scenario, its focus was to demonstrate how easily and quickly multiple users could model several pass-phrases of their choice. The UBM was built with data from either RSR or TIMIT, both databases gave similar results. The training procedure goes as follows:

1. Launch the demo in *Train Mode*
2. Select one slot for a given user and pass-phrase
3. Record 3 repetitions of the same pass-phrase (each recording can be listened to and saved or re-recorded), the model is now created
4. Create another model in a free slot or switch to *Test Mode*

Once at least one model has been trained, in test mode the user can record and test target-correct or target-wrong trials and —of course— with help from another person, impostor-correct and impostor-wrong trials (see Section II.5.2.b). The interface for testing is pictured in Fig. III.3 and is used as follows:

1. Select the slot correspondent to the claimed model
2. Record one utterance
3. Select "test" to obtain a score and a label
4. Select "add as target" or "add as impostor" depending on the ground truth, the appropriate symbol will be displayed on the score bar

Albeit far from a finished product, the usability and efficiency aspects of the 2-layer system were immediately obvious when, in all of the occasions the demo was shown, groups of people shown active interest in trying to fool the system (with little success) or see how close one's impostor scores were to the target scores of the other.

When testing offline like in the experiments reported above as well as in [59], the threshold for the EER is chosen a posteriori, a thing which is not at all possible when testing online. Although it is suggested to tune the threshold on the development set and use it on the evaluation set [37], this works well when the latter set exhibits similar if not identical recording conditions to the former set, which is the case with RSR2015 but not in real life. After a few experiments involving scores from RSR2015 data and recordings done internally at NXP, a fixed threshold value was empirically set to a log likelihood ratio of 2. Although this value was then confirmed to be optimal in several occasion in which the demo was presented (including noisy and reverberant environments such as poster sessions halls and corridors), it is clear to the author that the tuning process for the threshold was far from accurate. Therefore, it was left as a user-tunable parameter on the interface with a slider (see Fig. III.3).

It was eventually envisioned to automatically set the threshold as a function of the scores of the enrolment utterances against the model itself (*autoscores*), but no efficient solution was found. Said scores are, however, still displayed when a model is selected in test mode, as they might be interesting to compare with all four kinds of trial scores (TC, TW, IC, IW) at inference time (as can be seen in Fig. III.3, autoscores are predictably very high even when compared to the TC trial score, pictured as the first circle).

III.6 Conclusions

Short-utterance text-dependent speaker verification is the closest sub-field of ASV with the most potential to real commercial applications because of its reduced time and resources demands at inference time. Usability for the end user means non-intrusive interaction and plug-and-play setup, systems which demand minutes of the user speech to build a speaker model are not user-friendly. In automatic speaker verification for smart device/home applications and in the Internet of Things (IoT) domain, the collection of enrolment data is one of the most invasive and inconvenient tasks from the end user perspective. The objective of this work was to improve on the text-dependent HiLAM system usability by questioning the need for several minutes of the user's speech.

The experiments conducted with progressively subsampled protocols prove the non-necessity of the full text-independent enrolment used in the conventional HiLAM system, in the case that the ultimate recognition task is text-dependent in nature. It was found that reductions up to 66% of enrolment data do not influence the system performance. Results produced using a publicly available, standard database and protocols show that text-independent, middle-layer enrolment, while in some cases improving robustness slightly, unnecessarily impacts on usability. A simplified 2-layer system was implemented, speaker enrolment is then performed using only three repetitions of a given sentence or pass-phrase, reducing the training procedure asked of the user to just a few seconds. The proposed approach, admittedly a modest modification of the original system, delivers largely comparable levels of automatic speaker verification performance with a 97% reduction in enrolment data. The work shows that the middle layer of the HiLAM system and, hence, text-independent enrolment can be dispensed with entirely.

The resulting system relies on a few examples of the user's voice. As a short sentence is obviously phonetically scarce, its choice is understandably crucial. Even considering the standard HiLAM version, or any short-utterance text-dependent system, the question still holds: how much the choice of a given pass-phrase can influence the performance of such systems? Are there "strong" and "weak" pass-phrases? Can these trends be observed and generalised to everyone or are they impossible to meaningfully isolate from other factor like inter- and intra-speaker and session variations? These are all issues that any company that would like to set up a text-dependent authentication inevitably has to face. The chosen sentence, whether "my voice is my passport" or "ok Google", has to be the *one-size-fits-all* of automatic speaker verification. A study mostly concurrent with the one of this chapter, was carried out using the HiLAM system and the RSR2015 corpus and is the focus of the following chapter.

Chapter IV

Spoken password strength

In this chapter, a statistical analysis on the influence of the actual text content on the global performance of a text-dependent system is carried out; the objective is to identify trends related to the phonetic content of very short sentences and verify if those trends are consistent between different sets of speakers.

Our baseline was once again the standard 3-Layer HiLAM system described in Section II.3.5. The 2-layer variant implemented in the previous chapter was not used for this work because the starting point should be a standard system. Moreover, the lack of a text-independent speaker model in the simplified 2-layer system would undoubtedly increase the reliance on the text content, and result in observed behaviours may be just caused by our own modifications to the system.

The RSR2015 corpus is rather exhaustive in matter of combinations, as every speaker utters every sentence in the whole database, for the same amount of sessions. This fully-combined nature allowed for this study, done almost in parallel to the one in chapter III, in which the protocol was subsampled to (i) observe the isolated effect of the single sentences on the global performance of the system and (ii) find if such effect is consistent across different sets of speakers, in order to find *universally strong* or *weak* spoken passwords.

With a thorough statistical analysis, the work shows how significant reductions in error rates can be achieved by preventing the use of weak passwords and that improvements in performance are consistent across disjoint speaker subsets. The work this chapter is concerned with was first published in [5]. The ultimate goal is to develop an automated means of enforcing the use of stronger or more discriminant spoken passwords; a patent for this concept was filed and published [6].

IV.1 The concept of spoken password strength

Text-independent systems which adopt minute-long utterances as test trials usually assume the audio to be inherently normalised at the phoneme level. However, when the duration is within seconds length a good representation of all the phonemes in any language is just impossible. In this case, phonetic variation can have a significant impact on recognition performance [43, 60]. It is known that different speech units offer different levels of speaker discrimination: the work in [61], later extended in [62] analysed the idiosyncratic information contained in French vowels. While perhaps offering greater

insights relevant to the forensic branch of speaker recognition in terms of explaining results, the work points towards a mechanism for the selection or weighting of the most discriminant speech components for speaker modelling and recognition [63]. It is thus safe to assume that, just as is the case with written passwords, some spoken passwords or pass-phrases are more secure than others.

In an ASV system accept and reject decisions are made according to a *global threshold*. Whether this threshold is set a priori (see III.5) or a posteriori as in the case of a *global EER*, it always represents an inevitable compromise between the "inner" scores distributions related to an array of different factors, e.g. speaker-dependency, device-dependency and, in this case, text-dependency.

IV.2 Preliminary observations

Example target and impostor distributions with the global EER threshold depicted as a vertical green line are illustrated in the top row of Fig. IV.1: the threshold is set to give the same error rates for false positives and false negatives, whose value is directly proportional to the amount of overlap between the two distributions. In the case of the IC condition, the influence of text is quantifiable from the target and impostor score distributions for subsets of same-text trials. These distributions are referred to as *text-dependent distributions*. As illustrated in Fig. IV.1 for commands 35, 51 and 54 of the RSR2015 database there is thus a *text-dependent EER* obtained with a *text-dependent threshold* for each command. The text-dependent distributions are inner distributions to the global one; we can identify **two** factors that influence the global EER and are responsible for how big of a compromise the global threshold is.

IV.2.1 The text-dependent shift

The first factor is the relative "placement" of the scores: if both target and impostor score distributions for a given command are shifted higher or lower compared to most of the other text-dependent distributions, this will negatively affect global performance. This factor is not at all an index of strength or weakness for a given password because it is relative to other text-dependent scores; in fact, if there was just one sentence in the whole protocol the issue would be non-existent, regardless of the text content. This factor can eventually be mitigated with score normalisation techniques, in this case it would require cohorts of speakers uttering exactly the needed sentence, which is likely to be impracticable.

IV.2.2 The text-dependent overlap

The second factor is the overlap between text-dependent target and impostor score distributions, which is directly proportional to the text-dependent EER. Text-dependent distributions for the commands illustrated in Fig. IV.1 show 3 different scenarios when compared to the global distributions of the first row: command 35 has higher overlap and a higher threshold value; command 51 exhibits little overlap and a lower threshold value; finally, command 54, while having the closest threshold to the global one, shows a

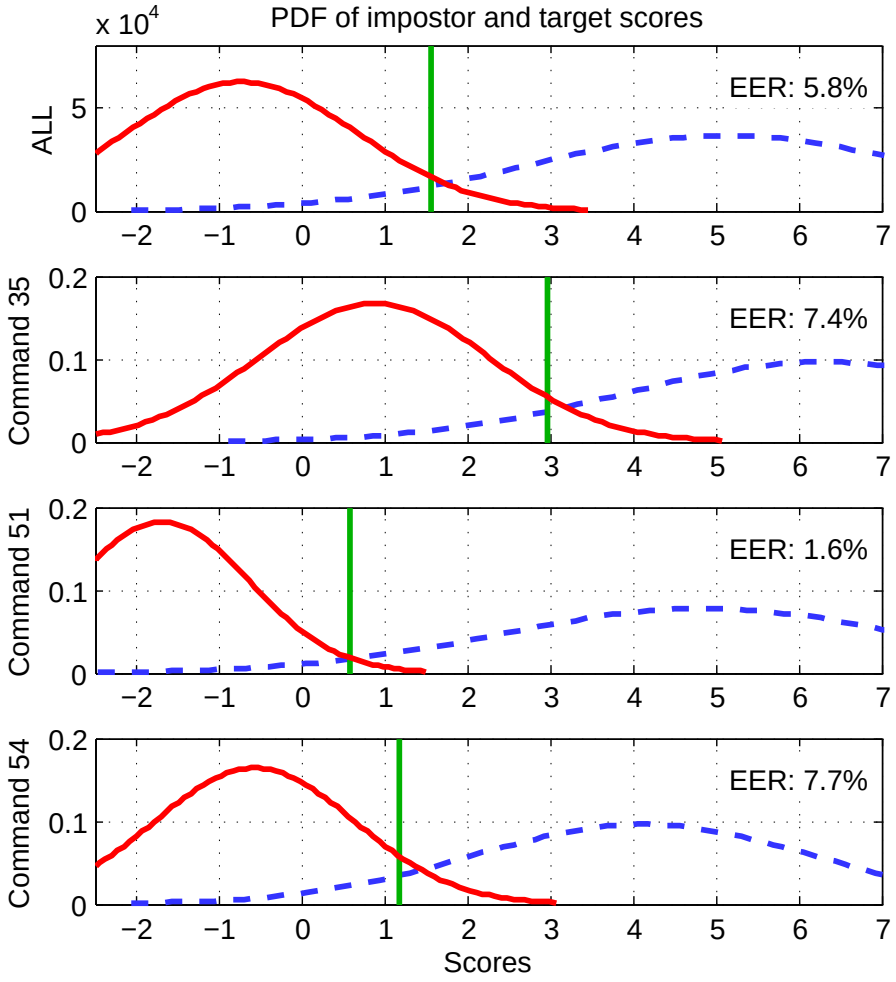


Fig. IV.1 *Impostor (solid) and target (dashed) score distributions and EER thresholds (vertical lines) from the male development set of the RSR2015 corpus. Plots illustrated separately for all commands trials (top) and for 3 command-specific trials*

considerable overlap between distributions which corresponds to the highest EER of the three.

These two factors contribute together to the global performance: to yield good results, each of the command-dependent score distributions must have a small target-impostor overlap area, and in addition the text-dependent thresholds must not be too far from one another, to make the global threshold less of a compromise. Nevertheless, only the overlap factor can considered as a measure of password strength: the corresponding text-dependent EER is the only insight to how difficult it is for the system to model and identify a speaker with respect to the chosen sentence. Some sentences are undoubtedly more user-representative than others across the whole set: the lower the text-dependent EER, the strongest the password.

IV.3 Database and protocols

This work was performed on part II of the RSR2015 corpus (male speakers only), namely *short commands* (see Fig. II.7). With an average recording duration of less than 1 second, it is the shortest-duration subset of the database [37], where the different phonetic content of each sentence supposedly matters more than in part I, which was used in the previous chapter. In order to assess variations in performance depending on the text content, experiments for this study concern only the IC (impostor-correct) condition, in which test trials and the claimed text-dependent model always share the same text content (see II.5.2.b).

When experimenting on Part II, Part I data is only used for the learning of background information and there is no overlap between speakers or phrases between the data used for background modelling and that used for training and testing. The sly impostor subset of Part II of the RSR2015 corpus contains 8990 TC (target) and 440510 IC (impostor) trials for the development set and 10250 TC and 574000 IC trials for the evaluation set. These numbers differ slightly from those reported in [35][1].

The baseline HiLAM system is, for the most, unchanged from the implementation described in chapter III. The only difference in the pipeline is in feature extraction. They are comprised of 18 coefficients (C0 removed) appended with deltas and double deltas for a total of 54 coefficients, compared to the 57 of previous chapter. This causes slightly worse performances for the IC condition, with EERs of 1.74% and 1.93% on the development and evaluation set, respectively (compare with Table VII.3). This is due to the fact that at the time when this work was done —and published— the HiLAM optimisation was still in progress. Since the conclusions derived from this study concern text-dependent variations in EER, a slight difference in performance of the baseline does not put this chapter findings into question.

IV.4 Statistical analysis

The following sections describe a statistical analysis that illustrates the potential to improve ASV performance through the selection of strong spoken sentences. It furthermore demonstrates that the notion of password strength is consistent across disjoint sets of speakers.

IV.4.1 Variable strength command groups

On the previously made assumption that a strong password is characterised by a relatively small text-dependent overlap, commands are first ranked by decreasing text-dependent EER. This ranking by strength is needed in order to simulate an eventual text-dependent ASV system that would include password strength recommendation.

This process is performed separately for the development and evaluation sets thus yielding **two** rankings for the same 30 commands. This step already showed similarities in ranking positions across the 2 sets and it is promising but not enough statistically significant to prove the concept of universal password strength.

[1] The authors became aware of the standard protocols for RSR 2015 Part II only after most of the work reported in this chapter was already completed.

From each of these rankings, groups of commands are formed by selecting 10 with the closest strength starting at every rank position, thereby producing 21 groups in total. The first group is comprised of the 10 weakest commands ranked #1 to #10, the second group is comprised of those ranked #2 to #11 and so on until the last group which contains the 10 strongest commands ranked #21 to #30.

IV.4.2 Sampling distribution of the EER

The significance of the difference in recognition performance obtained for each group is measured with the following bootstrapping procedure: for each of the 21 groups, 1000 populations[2] of 30 commands each are generated by picking at random from the 10 commands in the group, it is therefore very likely that a given command will be picked more than once (*i.e.* on a lower scale, a possible population of five resampled from a group two commands a and b can be $[a,b,a,a,b]$). This procedure is known as resampling with replacement [64]. This produces a total of $21 \times 1000 = 21000$ different testing protocols of progressively stronger passwords (according to the rank), whose size in terms of the number of trials is the same as that of the full RSR2015 protocols.

The random resampling of 10 commands to 30 is key to ensure statistical significance: instead of a single EER value per group of commands we now have a sampling distribution of 1000 EERs per group. From each of these distributions we derived a mean EER and a confidence interval. The sampling distributions were visually inspected for normality, allowing for 95% confidence intervals of 1.96 times the standard deviation of the distribution, thereby removing 2.5% of the observations at each end of the distribution. This interval around the mean EER of the distribution has a high probability of encompassing the true value of the EER for each group. Differences in performance obtained for groups with non-overlapping confidence intervals can hence be considered as being statistically significant.

Solid symbols in Fig. IV.2 represent mean EERs and confidence intervals of all the 21 groups for the development (a) and evaluation (b) sets. Another important goal was to see how well assumptions of password strength made for one set of speakers translated to the other. As explained in IV.4.1, two rankings were made according to the text-dependent EERs of each set. The unfilled symbols in Fig. IV.2 (a) and (b) correspond to EERs and confidence intervals obtained from groups of commands ranked according to a disjoint set of speakers, namely the evaluation-set-derived rank applied to development set trials and vice-versa. This is necessary in order to illustrate whether or not command strength is consistent across both sets.

IV.4.3 Isolating the influence of overlap

It is worth noting that, similarly to the original protocols which involved 30 —unique— commands, every population EER is a **global** EER which is still influenced by text-dependent shift and overlap factors explained in sections IV.2.1 and IV.2.2. This is why in Fig. IV.2(a) and (b) solid-symbol curves are not strictly monotonic, despite resulting from

[2]The term *population* can be misleading in this context, since it does not refer to speakers but instead to commands. The number of speakers is constant in all experiments and comprises the full development or evaluation set.

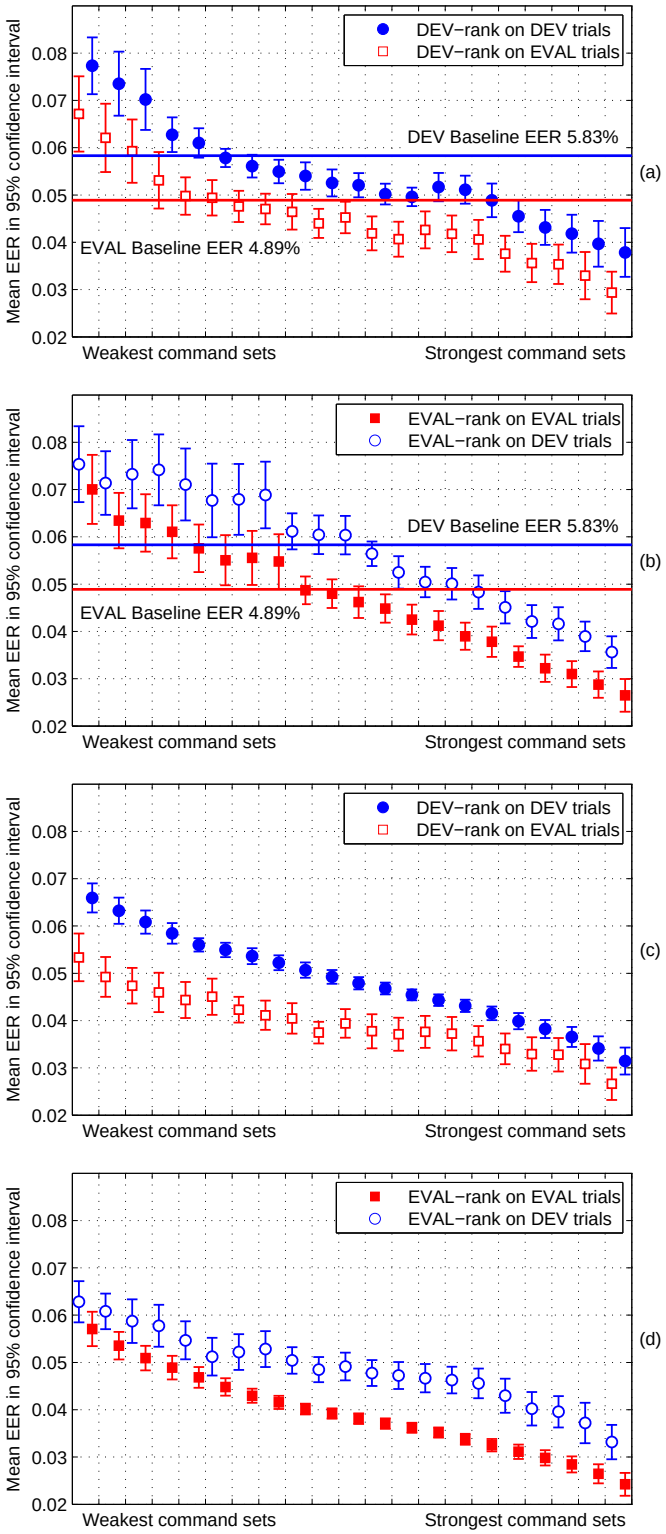


Fig. IV.2 *ASV performance with (c,d) and without (a,b) text-dependent threshold adjustment. Each point represents the mean EER over 1000 resamplings of 30 commands chosen with replacement among the 10 commands of each sub group. The horizontal lines in (a,b) represent the baseline performance of the system for both sets with all 30 commands.*

progressively stronger command trials. As the measure of a spoken password strength should not be influenced by scores unrelated to it, we can isolate the text-dependent overlap to observe only its influence on the system performance. To illustrate the dependence on overlap in isolation from threshold effects, the experiments described above are repeated with all trial scores normalised by subtracting the text-dependent threshold. The global EER for each command is then obtained with a score threshold of zero. This is equivalent to calculating the mean of all text-dependent EERs, each weighted by the number of trials concerning its command.

It is stressed that while this procedure obviously requires some "unfair" a posteriori knowledge, it was applied solely for the sake of observation of text-dependent behaviours and not to achieve lower error rates. Results for this experiment are reported in Fig. IV.2(c) and (d). As expected, the threshold adjustment renders the curves with ranking from the their own set strictly monotonic and confidence intervals smaller, since now they are just influenced by differences in resampling between populations.

IV.5 Results interpretation

When using their own ranking, EER results for both the development and evaluation sets show significant decreases as the group contains increasingly stronger commands – solid-symbol plots in Fig. IV.2(a) and IV.2(b). When using threshold-adjusted scores (solid-symbol plots in Fig. IV.2(c) and IV.2(d), decreases are strictly monotonic. This observation confirms that the spread of text-dependent thresholds also affects performance.

Other observations concern results for cross-set rankings – unfilled-symbol plots in Fig. IV.2(c) and IV.2(d). Rankings made on the development set translate well to the evaluation set and vice-versa. For the evaluation set, results illustrated in Fig. IV.2(a) show that only 6 groups have an EER which is not significantly different to the overall EER (4.89%). For the development set, results illustrated in Fig. IV.2(b) show only 4 groups with a non-significantly different overall EER (5.83%). The significant global decrease in EER (albeit non-monotonic) shows that, with negligible differences in ranking, some commands are consistently 'weak' across different speakers. According to these results, a system including a password strength acceptance criterion could halve the error rate by choosing (multiple) stronger sentences over weaker ones (from 5.34% to 2.67% on the development set, and from 6.28% to 3.32% on the evaluation set). Finally, we note that the visible offset of the evaluation set EERs is inherent to the RSR2015 database and consistent with results presented by others [35, 59].

Although in this study it was not possible to track exactly which specific phonetic content was responsible for a weak or a strong password, some intuitive, high-level observations are nonetheless offered. Consistent to both development and evaluation sets is the higher ranking of longer duration sentences. This is not surprising. Other observations are more intriguing. While commands such as 'Turn on light', 'Watch Cartoon' and 'Volume Down', all of similar duration, all perform well across both subsets, others of similar length such as 'Door Open', 'Volume up' and 'Aircon off' performed poorly across both subsets. Given the similar duration, it is assumed that the first three commands have more discriminative phonetic content. 'Volume up' and 'Volume down' vary only by the last two phonemes but are ranked among the weakest and strongest commands

respectively. These observations are consistent with the discriminative power of nasal sounds studied in [63]. Clearly these factors warrant further attention in future work.

IV.6 Conclusions

When dealing with short pass-phrases or passwords a few seconds length, the text-content influence matters as much as inter-speaker or inter-handset variations. In this context, the authentication strength of a text-dependent model very much depends on the uttered sentence. The work in this chapter investigates the influence of text content on password strength with a thorough statistical analysis. Two independent sets of speakers were tested on very short commands trials, yielding considerable differences in text-dependent EERs. It was demonstrated that the ranking of commands from weak to strong according to their impact on system performance can be assumed from one set of speakers and applied to another within statistical significance, proving the concept of spoken password strength is consistent.

After the findings of this work were published in [5], research continued in order to investigate the exact speech patterns that caused a spoken password to be universally weak. The ultimate goal was to develop automatic means of identifying weaker passwords to give the user immediate feedback on unsafe choices. This a priori knowledge could have also been used by companies while choosing the branded wake-up pass-phrase that offers higher level of discrimination among different speakers. However, due to the scarcity of phoneme-level labelled data, no significant progress was made in the implementation of an actual system. Nevertheless, a concept design of the system itself, with a detailed action flow was described and published in a patent [6].

Chapter V

A review of deep learning speaker verification approaches

The literature review in chapter II covers the most popular system and techniques used in "traditional" ASV prior to the *deep learning* revolution, often built upon previous approaches, and some core elements —such as MFCC extraction— remained standard for decades. This chapter describes the neural network "takeover" in the speaker verification domain that has been under way for some years. This has today come to the point where some of the underlying assumptions and core techniques common to all ASV systems seem sub-optimal in the face of deep learning approaches.

Deep neural networks were first applied to image recognition [65] and their first applications to ASV often concerned only a block of the entire pipeline, either in the front- or back-end. Later, ASV approaches became entirely DNN-based, and completely rewrote the ASV paradigm in so-called end-to-end approaches. Understanding the direction in which the latest ASV research is going, its current limits and assumptions is of key importance to the contributions in Chapters VI and VII, as they push the boundary of machine learning beyond hand-crafted features and fixed, layered network architectures.

V.1 Neural networks and deep learning

This section briefly introduces the main neural network families, most of which first succeeded in domains other than ASV and audio in general. The terms *deep neural network* (DNN) and *deep learning* (DL) are often treated as inseparable, but one does not necessarily imply the other. It is a common acceptation that, to be called "deep", a neural network must have two or more hidden layers. Deep learning (although it can be applied to non-NN approaches, termed as *unconventional deep learning* [66]), can be seen as the ensemble of algorithms, techniques and approaches that made deep neural network training possible at first and wide-spread later.

Regular *feed-forward* networks (no recurrent paths) with one hidden layer are referred to as *shallow* networks. The *universal approximation* theorem [67] states that a shallow network is, in principle, capable of representing any continuous function. Nevertheless it was observed empirically that deeper networks led to better performance in many applications [68]. The reasons for the correlation between depth and performance are

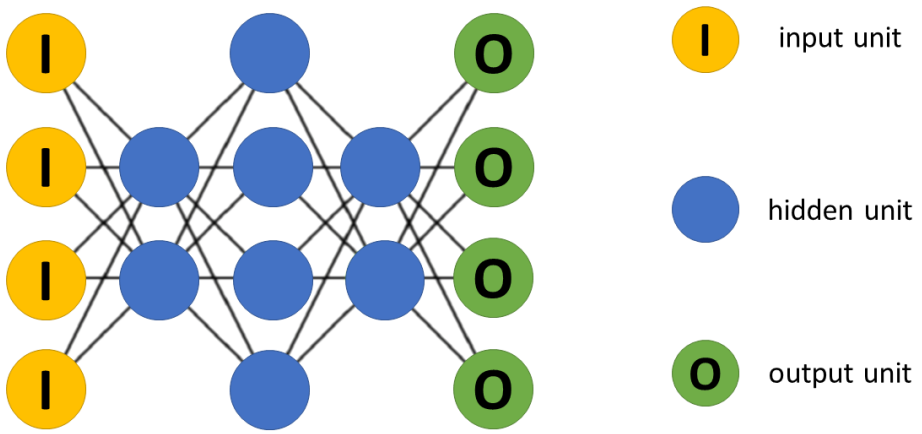


Fig. V.1 *A deep belief network, characterised by multiple hidden layers with no lateral connections.*

believed to be several: from the fact that a shallow network would need a considerably higher number of neurons compared to a deep network to achieve the same results, to the inherent hierarchical nature of the tasks. Empirical evidence shows that deep networks perform better than shallow networks in real world tasks, where hierarchical knowledge is often involved (i.e. pixels, edges, shapes and objects for image recognition) [69]. As a result, state-of-the art NNs became increasingly deeper and, even if studies questioned the real need for such depth [70, 71] as of today, there is no clear answer on "what makes them work", with very deep networks being the norm. What was clear since the late 80s was that training DNNs with the then-standard *back propagation* algorithm was not feasible [65], with one of the most hindering factors being the *vanishing or exploding gradient* problem.

The first milestone responsible for deep learning resurgence in the last decade was the introduction of *unsupervised pre-training* in 2006 [72], which started a chain of innovations, from *Hessian-free optimisation* [73] to enhancing *stochastic gradient descent* [74], to *dropout* [75] to *residual networks* [76]. While explaining those contributions is far beyond the scope of this thesis, they really show how progress in this field is still very rooted in trial and error as opposed to truly understanding the reason of each improvement.

V.1.1 Deep Belief Networks

Deep Belief Networks (DBN), introduced in [72] are generative models composed of multiple interconnected "isolated" hidden layers, as the units from one layer are only connected to units belonging to the previous and next layers, with no lateral connections (see Fig. V.1). The weights of the connections are optimised via "greedy" layer-by-layer training: it starts with the first layer as the (observable) training set, used as input for

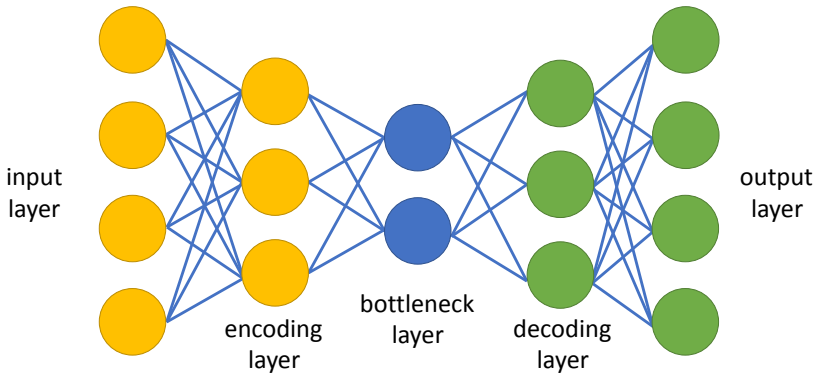


Fig. V.2 *Structure of an auto-encoder. Deep auto-encoders usually present more encoding and decoding layers.*

the first hidden layer, which in turn feeds its output as the input to the subsequent layer, allowing for fast unsupervised training. This step can be followed by a fine-tuning supervised step, in which labelled data is fed to the network and the error derivatives are then back-propagated.

The DBN hidden units are stochastic latent variables with their output being a weighted sum of the inputs passed through an activation function, often of the sigmoid or rectified linear type. DBNs can be seen as a stack of NNs with a single hidden layer, each of which "sees" the output of the previous layer as observable data and learns to output features that represent higher-order correlations in the data [77]. The breaking-down of bigger problems into simpler, smaller ones is what made deep learning on DNNs with millions of parameters actually feasible.

In general for DNN approaches, the computational effort at inference time is considerably lower than at training. Nevertheless, if the goal is to test the network with almost no latency on an embedded DSP chip, things such as the number of connections in the order of hundreds of thousands, units connected with non-linear activations and high bit-depth signals are still prohibitive.

V.1.2 Deep Auto-encoders

The principle behind auto-encoders (AEs) is to map the input patterns to themselves by passing through a simpler representation, in terms of parameters. This is achieved by having one hidden *bottleneck* layer, narrower than the preceding ones. This forces the network to learn a compact representation of the input (often referred to as bottleneck features), which is then followed by decoding layers which "decompress" the representation to be as close as possible to the input, in an analogy to data compression algorithms (see Fig. V.2).

Deep auto-encoders (DAEs), also referred as auto-encoder stacks are (as the name suggests) composed by several stacked AE networks, with the bottleneck layer of one AE

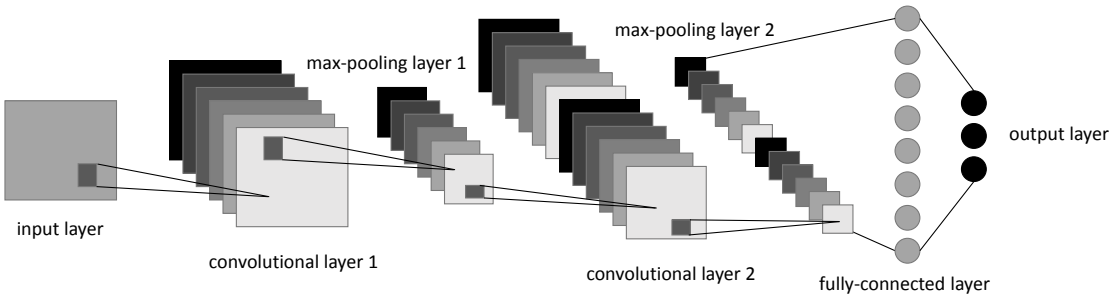


Fig. V.3 *A typical CNN architecture: convolutional layers are interleaved with max-pooling layers, the former capture information (i.e. edges), the latter prune the information to keep what is most relevant*

acting as the input layer for the next (narrower) encoding layer, followed by a symmetrical decoding structure. These are trained in a similar way to DBNs, with a first pass of unsupervised pre-training followed by supervised tuning via back-propagation [65].

V.1.3 Convolutional Neural Networks

Convolutional Neural Networks, (CNNs) are feed-forward networks that are able to extract progressively higher level features from the input, such as 2-dimensional arrays of pixels, hence their popularity in image recognition [65]. The output of a convolutional layer is calculated by applying (one or more) convolution filter(s) on the input and shifting it with a fixed step across either a section (*receptive field*) or the whole input array (*fully connected*). The resulting output retains the same dimensionality of the input and is fed to the next layer.

To reduce the dimensionality and capture different orders of spatial variations the filter *stride* (the size of the shift step) can be incremented, eventually compensating with a larger receptive field (which will bring additional computations). Some layers have the explicit purpose of dimensionality reduction: *subsampling* layers reduce the input of neighbouring neurons to a single output, either via averaging or *max pooling* (see Fig. V.3). CNNs are among the most resource-demanding deep learning approaches even at inference, making them the least preferred candidate for embedded solutions. They have greatly benefited from their implementation on graphic processing units (GPUs), which considerably speed up the whole training process and allow for human-competitive performance in image classification [78].

V.1.4 Long short-term Memory Recurrent Neural Networks

Any kind of NN architecture that exhibits a cyclical path is classified as a recurrent neural network (RNN). RNNs are inherently capable of retaining past information and are therefore well suited to tasks involving the modelling or classification of dynamic

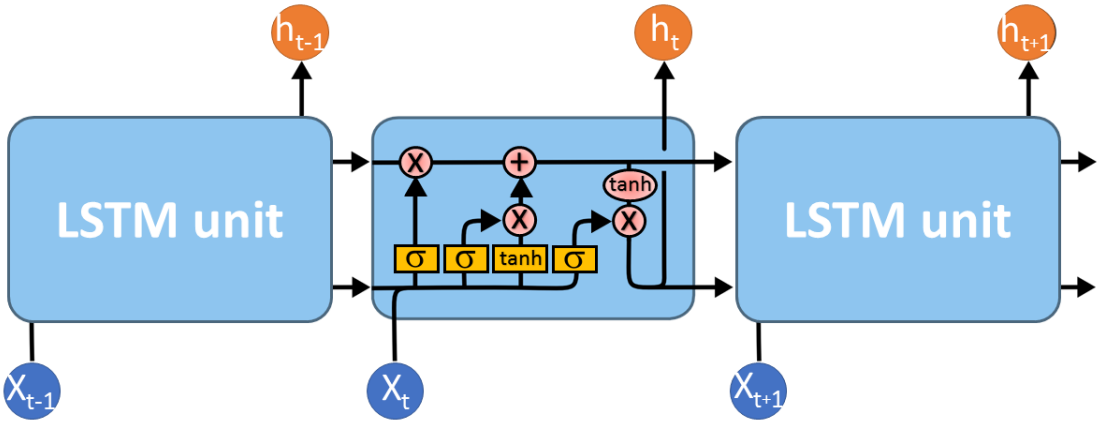


Fig. V.4 *An LSTM architecture unrolled in time. Each unit contains 4 interactive layers, depicted as yellow rectangles. These are responsible for which past values are forgot, kept, updated and output*

or time-variable data. Nevertheless, deep RNNs have proved difficult to train with back-propagation through more than a few time steps, plagued by the aforementioned exploding gradient issue. The solution and now state-of-the art architecture for RNNs came in the form of *long short-term memory* (LSTM), introduced in [79].

LSTMs are capable of learning long-term dependencies; they are immune from vanishing or exploding gradients and do not need unsupervised pre-training. This is achieved through memory units which include *constant error carousels* whose error derivatives of 1.0 cannot explode nor vanish. Other non-linear units are responsible for keeping or discarding current and past information: they are called *forget-*, *keep-*, *update-* and *output-gate*, pictured in this order in Fig. V.4. LSTMs can easily take into account information that occurred thousands of time steps prior. Most successful RNN approaches are based on LSTMs.

Recurrence is costly to implement on very limited-resource hardware. In the case of feed-forward architectures, to save memory, the unit values of a given layer are usually discarded after one or a few steps, because they are only needed to calculate the unit values of the subsequent layer and their memory locations can be reused right after. A fully connected network could take up just around twice the storage space of the unit values of the largest layer, but with every recurrent connection an additional value must be stored, considerably increasing memory usage.

V.2 Deep learning in ASV

In recent years, deep learning has entered the ASV field in various ways. The neural network architectures described above are often used to replace specific elements of the

speaker verification toolchain explained in Chapter II. This section describes the ways in which different blocks of the ASV pipeline have benefited from being "enhanced" with deep learning techniques.

V.2.1 Feature extraction

As highlighted in sections V.1.2 and V.1.3, deep neural networks can be applied to dimensionality reduction, opening up the possibilities of using them for speaker feature extraction. While traditional acoustic features rely on human knowledge, leaving the task to machine learning is undoubtedly a step forward, because different features could be extracted for different tasks, or even different speakers. The potential was first proved in speech recognition tasks [80].

The DNN architecture used in [81] works on stacked Mel-filterbank energies and uses the activations of the last DNN hidden layer as speaker-specific frame-level features for text-dependent ASV. Utterance-level representations are obtained by averaging frame-level features thereby giving so-called *d-vectors*. At test time, the score is simply computed as the cosine distance between the enrolment and test utterance d-vectors. Recent work [82] on text-independent ASV achieved feature learning with less dependence on the back-end (the focus was on learning truly discriminative features that work well with any back-end). The solution adopted a convolutional *time-delay* DNN with a bottleneck layer (see Section V.1.2) which achieved remarkable performance on extremely short durations: with just 3 seconds of speech for enrolment and 0.3-second test utterances, the proposed system achieved EERs as low as 14% on the Fisher dataset, whereas none of the baseline i-vector system configurations achieved EERs of below 33%.

V.2.2 Applications to i-vector frameworks

The use of DNNs for the estimation of hidden Markov model state posterior probabilities [83,84] in speech recognition led to their use for the estimation of phonetically-aware frame posteriors, replacing the UBM i-vector frameworks, for both text-dependent [85] and independent [86] systems. In these cases deep learning techniques are applied to extract i-vectors, as well as to add temporal sensitivity and perform frame alignment. The DNN architectures used for these works were in fact trained for speech recognition.

DNNs also influenced i-vector-based systems from another angle: in [87] DNN embeddings were used as an alternative to i-vectors within a probabilistic linear discriminant analysis (PLDA) framework. The fixed-dimension speaker embeddings were computed from one of the last hidden layers of a feed-forward DNN trained in a text-dependent manner. At test time, pairs of embeddings are scored using a PLDA-based backend as in a conventional i-vector approach (see Section II.3.4). Experiments for the system in [87] on NIST SRE16 data (see II.5.1.d) reported lower language-pooled EERs for an embeddings-based system (11.9%) versus i-vectors (13.6%).

V.2.3 Back-ends and classifiers

The conventional classifier architecture of a DNN is trained on stacks of short-term features with a temporal context of several preceding and following frames. The output units of the DNN are the prediction of the posterior probabilities for each class, whether

phonemes or closed-set speaker IDs [88]. With speaker verification being a binary classification task, the network is trained to recognise only one target speaker, with the possible outputs being only the positive and the alternative (background/impostor) hypothesis [81].

In a somewhat peculiar example of the use of deep learning in a speaker verification back-end [89], an auto-encoder is trained to "mimic" the discriminant analysis procedure and then is effectively used in lieu of PLDA scoring (see Section II.3.4) in the system pipeline. When compared with "regular" PLDA scoring on the same i-vector framework, the proposed approach yields 36% relative improvement on NIST SRE10 data.

V.3 End-to-end

Up until recently, feature extraction, modelling and classification have been considered separate, standalone blocks of the ASV pipeline. The advent of deep learning and neural networks allowed for approaches where the parameters of each block of the whole system are adjusted in one joint effort. These jointly-optimised systems are referred to as *end-to-end* (E2E) systems. As deep learning "invaded" the ASV field, the last two years saw the end-to-end paradigm applied to speaker recognition [90–92].

Traditionally, certain feature extraction aspects (*i.e.* the assumption of frame independence) were tailored to specific modelling techniques such as GMM and, in turn, the model was built with the classifier in mind. It could be assumed that any E2E system leaves all previously hand-crafted parameter optimisation to the machine learning algorithm. However, there are two aspects often both present in most current end-to-end approaches that still rely on pre-determined and engineered choices: middle-level representations (*i.e* spectrum, cepstrum) and fixed topologies, which are explained below.

V.3.1 Middle-level representations VS raw audio

Spectrograms and other middle-level representations are still the input signal of most ASV deep learning approaches and even end-to-end systems. In the latter cases this clashes with the end-to-end name because the input is somehow still hand-crafted, making the pipeline not truly end-to-end. Feature-learning focused work in [81] uses 40 mel filterbanks as they were found to be better suited to NN acoustic modelling than MFCCs [93]. Nevertheless, the work in [91], which bears the "end-to-end" moniker in its title, still feeds the system with traditional MFCC features.

The 2-d spectrogram (see Section II.2.2) seems to be a common choice for end-to-end systems [92, 94]. Treating audio sources as images is indeed handy, considering the first successful uses of deep architectures belong to the image recognition domain [65], allowing for an easier adaptation of an image recognition system to audio classification. Nevertheless, while an image is a raw (albeit discrete) representation of light, spectrograms are used as a form of pre-processing. Spectral representations stem from a linear transformation of the raw waveform, but they still rely upon the frame-blocking of speech signals into fixed-length windows. Even when the neural architecture is tailored to exploit speech dynamics [95], the input is often a reduced-dimensionality representation of spectral magnitude information. Usually, phase information is discarded entirely and assumptions of short-term stationarity and pseudo-independence between adjacent frames are made. In

doing so, the choice of what information to discard is not left to the system but instead predetermined.

The shift from cepstral features to filterbanks to spectrograms depicts a trends that goes towards less hand-crafted, less processed input data, at the end of which there is the **raw audio waveform**. Applying deep learning directly on the raw audio waveform is not infeasible nor inconvenient, as is testified by related work in areas neighbouring speaker verification *i.e.* speech [96] and emotion [97] recognition, in addition to spoofing detection [98].

A couple of examples of raw-audio speech recognition show how traditional, engineered speech processing aspects can be confirmed and/or refined. In [99], where the NN is fed raw audio, it was observed that the first layer hidden units learn impulse responses very similar to those of *gammatone filters*. When sorted by the fundamental frequency to which they responded, higher-frequency units corresponded to higher bandwidths, somewhat confirming the principle behind the Mel scale. Another speech recognition system, described in [100, 101] utilises time convolution to learn a so-called *brainogram*, a 2-d representation which gave better performance than the hand-crafted Mel-filterbanks, although with the same dimensionality.

The raw waveform has also been used to train speech synthesisers for text-to-speech: the work in [102] makes use of raw audio as input to learn cepstrum coefficients and then inverse-filters them to obtain the output speech. The speech synthesiser in [103] applies *dilated causal convolution* directly on audio samples to learn a generative model of speech. Avoiding dependence on hand-crafted features is clearly among the interests of DNN-related research.

Until recently, there was no end-to-end system trained directly on raw audio waveforms for automatic speaker verification. Only in the last year has the raw audio signal paradigm made its debut in ASV. Among the publications [104, 105] there is the author's own work [8] which is the focus of Chapter VI.

One other, recent example is SINCNET [106], which, while in fact limiting the first convolutional layer of a CNN to learn only the bandwidth of *sinc* filters, actually helps the network to transform raw audio into a representation which is meaningful to both humans and the back-end. While the concept of band-pass filters is clearly more human-friendly than a large number of filter parameters, its helpfulness on the back-end side is proven by the fact that SINCNET outperforms other CNNs architectures (fed with raw audio or mel-filterbanks) and DNNs fed with MFCCs. Experiments on the *Librispeech* datasets show that the EER loss brought using sinc filters for raw audio (-0.04%) is more significant than the loss brought by using raw audio versus mel-filterbanks (-0.01%).

V.3.2 Fixed topologies

Although having jointly optimised blocks inside a large neural structure, end-to-end architectures still exhibit fixed topologies with given-purpose modules responsible for different tasks. The networks present a layered, hierarchical structure [90, 107] in which the number of layers, their connectivity (local or full), the number of units per layer and their activation function (linear, rectified, *etc.*) are all predetermined. These choices preventively limit the search space which will be explored during the training phase, with no way of knowing if the topology is optimal for the task.

As with hand-crafted features, the will to leave the choice of the structure to the learning algorithm is present in the literature, although beyond the field of speech processing: one notable example pertaining to image recognition is [108], in which *reinforcement learning* is used to learn parameters of the network that relate to its structure such as the number of layers of a certain type (convolutional, fully connected, pooling), their width in terms of units, the number and size of receptive fields (see Section V.1.3), etc. Albeit considerably more versatile and allowing for thousands of possible architectures, these are still organised in a hierarchical fashion and connections are shared only between adjacent layers.

Residual networks (ResNets) [76] were introduced to compensate for training degradation caused by the addition of more layers. The key aspect of a ResNet is the use of *shortcut connections* to pass the information (often with an identity function) between non-adjacent layers. This hierarchy-breaking technique can be interpreted as a sign that conventional deep architectures may be sub-optimal.

Less-constrained solutions to design network architectures were proposed decades prior to the aforementioned work [109]. These often rely on *genetic algorithms* and *evolutionary strategies*, the focus of Chapter VI.

V.4 Summary

This Chapter describes the main neural network based approaches that are relevant to deep learning in ASV. While the domain of the first successes of deep learning and neural networks in general was image recognition, it is shown how, at first, deep neural networks were used to enhance one or more independent parts of the traditional ASV pipeline, to the point where every block of the toolchain employed deep-learning; the latest trend being jointly optimised, end-to-end approaches. Weaknesses in the current E2E paradigm are then criticised, because of the reliance on hand-crafted representations of the source audio and its use of complex, hierarchically layered fixed structures.

An ideal end-to-end system would have raw speech input on one end and the target/impostor decision outputs or scores at the other end. The entire structure should be interconnected with no explicit task tied to a specific part of the network. If such a system could provide reliable speaker authentication as well as more rather clear insights into how the network arrives to the decisions/scores it produces, that would truly be the ultimate end-to-end system. An experimental end-to-end ASV system working on raw audio with a non-fixed topology (first published in [8]) is presented in Chapter VI.

Chapter VI

Augmenting topologies applied to ASV

Neural networks are nowadays at the forefront of ASV research. Their complex structures make them resource-intensive even at inference, which in turns makes these approaches difficult to run on embedded devices, often requiring to downgrade the architecture. Chapter V describes the trend of deep learning approaches to leave progressively more freedom to machine learning, *i.e.* moving away from engineered MFCC features towards more descriptive inputs such as spectrograms. 2018 saw the introduction of raw audio as an input for ASV systems [8, 104–106]. It is highly likely that, in the near future, all systems that bear the "end-to-end" moniker will operate on raw audio inputs.

As explained in Chapter I, companies which already sell voice-enabled services usually run the whole process in the cloud, sending the recording from the device and receiving the feedback from a more powerful, distributed computational architecture. NXP, as a semiconductor company and embedded software developer, has interest in offering low-latency, lightweight processing operating directly on smart devices. Keeping users' private data on the device brings added value to products, especially after the recent introduction of the European *general data protection regulation* (GDPR), which limits the ways in which biometric data can be collected, transmitted and exploited.

With the progressive interest in raw audio by the research community and the problems posed by complex hierarchical structures, the research path of this thesis took a very experimental turn as the author started investigating evolutionary approaches. At the time of writing, evolutionary strategies have been around for more than two decades and rely on genetic algorithms: when applied to neural networks, instead of just one structure, a whole population of networks is evolved through several generations. Of major interest are the evolutionary approaches that include among the evolution parameters the architecture of the network; they have the potential to reach the performance of competing deep learning approaches with a network architecture that is tailored to the task and is often less complex in terms of the number of connections.

This chapter describes work (published in [8]) concerning the first application of raw audio **and** evolutionary, small-footprint topologies to ASV, and is organised as follows: Section VI.1 introduces the general topic of evolutionary strategies, and then goes progressively more into detail (and sub-categories) by diving into NeuroEvolution, Topology and Weight Evolving Neural Networks (TWEANNs) VI.1.1 and the NeuroEvolution of Augmenting Topologies (NEAT) algorithm VI.1.2. The latter is the basis of NXP work on raw audio classification, described in section VI.2, as well as for the contribution of

this Chapter, the application of NEAT to ASV, explained in Section VI.3.

Experiments for the proposed and baseline systems on the proprietary NXP database are described in Section VI.4, additional experiments on a subset of the NIST SRE16 dataset are described in Section VI.5, where the proposed approach is also compared with a system submitted to the 2016 NIST SRE. Conclusions are reported in Section VI.6.

VI.1 Evolutionary strategies

The core aspect of any evolutionary-based technique, and one which sets it apart from every other machine learning approach discussed in this thesis, is its reliance on genetic algorithms to search the solution space. Evolutionary strategies do not modify parameters of the system by back-propagating the error derivatives, as is the case with conventional neural network architectures and gradient descent-based optimisation. Instead of a single architecture, a population of potential solutions is adopted, each individual being defined by a unique set of parameters. The population is then evaluated by means of a fitness function, and the fittest individuals act as the basis to generate a new population, in a biological analogy to *natural selection* [110]. The way in which the new generation "inherits" aspects from the previous generation can be probabilistic, for example in genetic algorithms such as the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) [111], the new population is sampled from a multivariate Gaussian Distribution. At each generation, the means of the distribution are updated according to the fitness of the best individuals.

NeuroEvolution [112] is a form of evolutionary strategy that goes even closer to natural evolution when it comes to generating new individuals: parameter values of the next generation are obtained through biological analogies of mutations and crossovers between networks. These occur randomly (within set boundaries) during the evolutionary process. In what is called conventional NeuroEvolution (CNE), weights are evolved while the architecture remains fixed. Approaches which evolve the architecture along with the weights are explained in the next section.

VI.1.1 TWEANNs

One researching alternative to large neural architectures explained in Section V.3.2 is structure-evolving approaches known as Topology and Weight Evolving Neural Networks (TWEANNs). TWEANNs evolve the structure of networks along with the weights, by changing, adding or removing nodes and connections. Being a NeuroEvolution approach, TWEANNs produce new individuals by making networks "mate". An illustrative example of a single generation cycle is depicted in Fig VI.1. The process goes as follows: (i) the whole population is evaluated with a fitness function on the training data; (ii) the fittest individuals are selected (iii); the selected individuals generate a new population by either mutating or mating and producing offspring. The cycle is repeated until one individual of the current population either resolves the task, yields sufficient fitness or until any other suitable stopping condition is met.

One of the key aspects shared by NeuroEvolution approaches is the search for novelty through diversity. The optimisation process is not based on gradient descent and back-propagation, but instead consists in evaluating each network in the population on the

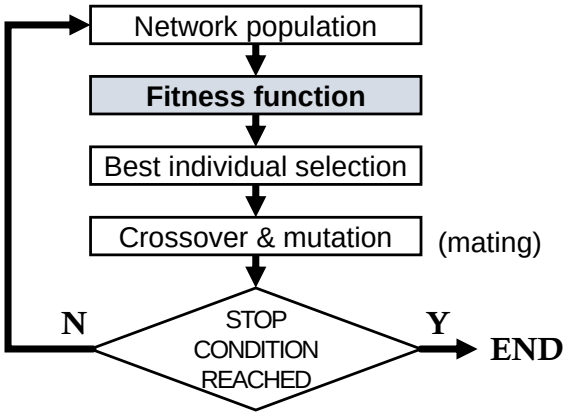


Fig. VI.1 *An illustration of one iteration of evolution: the performance of each network in a population is assessed by the means of a fitness function. The* ***best*** *individuals are selected to form a new generation of networks.*

training data according to a fitness function, in the same manner as would be done during inference. New connections can be created within any pair of nodes; this not only allows for recurrent paths, but completely abandons the notion of hierarchical layers. This also means that TWEANNs are not affected by the *vanishing/exploding gradient* problem mentioned in Section V.1 which is caused by the error derivatives being propagated too many times from one layer to the next, to the point where they either disappear or become too large. In architectures such as ResNet [113] this issue is mitigated by creating connections between non-adjacent layers; in TWEANNs there is no need to apply patches, as these connections are created automatically, by design.

Including the topology in the roster of optimisation parameters augments the space of solutions considerably. The key aspects of genetic algorithms, and in turn natural genetics, are what makes TWEANNs fascinating: pursuing novelty instead of pure improvement, conceiving randomness and diversity as tools to "cut corners" and to eventually find a better, simpler solution. Nevertheless, dealing with an evolving population of networks raises several questions:

- *What is the structure of the first generation of networks?*

  In many TWEANN approaches, "generation zero" is made of randomly generated architectures [114]. However, while as a genetic principle, diversity in the pool of individuals is highly desirable, initialising the population with random parameters often results in networks which have ineffective connections or that are just overall inefficient but that will nonetheless "survive" for several generations.

- *How should units and connections without layers be described to ensure unique networks?*

  With another biological analogy, evolving networks have a *genotype* and a *phenotype*: the former is akin to DNA and contains all the information about the network (including what could be needed when mating); the latter is the network graph, literally "what it looks like". The genotype can be indirectly or directly encoded.

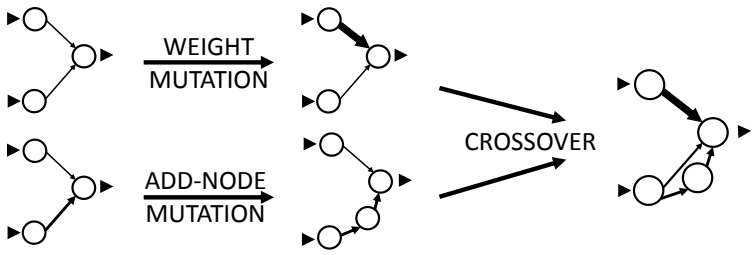


Fig. VI.2 *Mutation of weight (here symbolized by connection thickness), node adding and crossover: the three forms of network evolution.*

Indirect encoding involves high level rules on how a given network could grow or mutate, similarly to a decision tree. Without explicitly coding every node and connection, the genotype has to be derived from the phenotype, this could bias the evolution in unpredictable ways [114]. Direct encoding, on the other hand, while being less compact, maps every gene to a particular aspect of a given architecture (*e.g.* connection from unit a to unit b with weight c), with the phenotype being the visual counterpart of these instructions.

- *When two networks mate, what are the rules governing the generation of offspring?*

  A known issue with TWEANNs is *the permutations problem* [115]: at one point in evolution, two networks which exhibit a part of their structures which is responsible for a given function may loose that functionality when mating and be left with redundant parts. If the genotype does not contain any indication of what are — allowing another biological metaphor — the *eyes* or the *hands* of a network, *damaged offspring* may be the result [114].

VI.1.2 NEAT

NeuroEvolution of Aumenting Topologies (NEAT) is a take on TWEANNs that not only fulfills the requirements for low-footprint networks, but also addresses in an elegant way the aforementioned issues involved with their use, *e.g.* the permutations problem. When NEAT was introduced in 2002 [114], neural networks and deep learning were a few years away from the resurgence the field experienced in the 2010s that is described in Chapter V.

One crucial aspect of the NEAT algorithm centres around the incremental evolution of structure: the first generation is populated by bare-bone, minimal structures. Topologies are augmented iteratively through the addition of new nodes and connections thereby following a *complexifying* principle. Even if the algorithm does not incorporate an explicit measure of complexity, networks tend to remain comparatively simple in structure compared to typical deep neural network solutions. Possible mutations range from a simple weight change to the creation of new connections and nodes, to a crossover between two parent structures; these are illustrated in Fig. VI.2.

NEAT is remarkable for having provided elegant solutions to several problems that affected research in the field for a decade prior to its introduction:

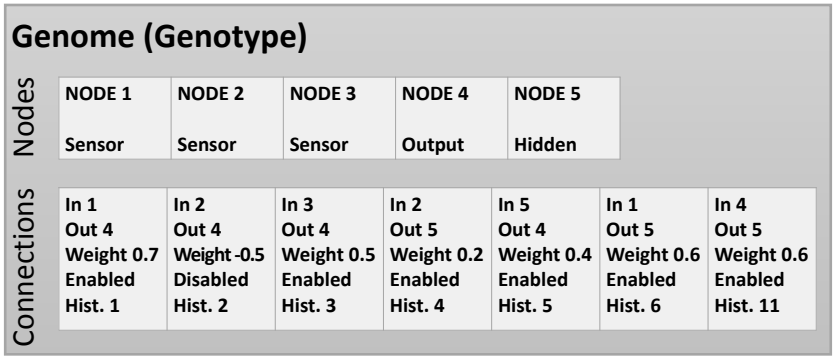


Fig. VI.3 *A NEAT genotype is a direct and self-contained textual representation of a unique network, which contains (as in nature) more information than that which can be observed in the resulting structure. Figure reproduced with permission from [114].*

- **Direct encoding**

  Taking another page out of natural genetics, NEAT makes uses of genotype direct encoding (see Fig. VI.3). A NEAT genotype specifies all nodes and connections of the network (each described in its specific gene) and contains all the information needed to uniquely identify a network. The visual representation of the architecture (the phenotype) can be inferred from the genotype but only the latter is necessary and sufficient during the whole evolutionary process.

- **Historical markings**

  Continuing the analogy with natural genetics, the genotype contains crucial information which cannot be "seen" from the phenotype. One of these is the embedding of *historical markings* in the genotype. Every time a new connection occurs, it represents a new branch of evolution: a global *innovation number* is then increased and assigned to the new genes, so that all branches that started from the same root share the same series of historical markings. Genes with the same innovation number are considered to be of the same trait (*i.e.* the eye colour in human DNA). This is of critical importance when two networks crossover: genes align with respect to their innovation numbers, the choice of inheriting gene x from one parent network, or y from the other, is possible only if both genes belong to the same trait as illustrated in Fig. VI.4.

- **Speciation**

  A direct consequence of having to perform gene alignment is that at some point in evolution, a pair of networks could have several genes which do not align because the corresponding traits are not present in both. These are called *disjoint genes*. The number of disjoint genes is used as a measure of distance between two networks: over a certain threshold, the networks are deemed incompatible and they are classified as belonging to different *species*. *Speciation* is what ensures diversity among the structures in the population, and allows networks to compete within niches instead

of one global pool (*i.e.* by selecting the fittest networks within the same species). This fuels innovation, as structural mutations that might lead to better performing networks in later generations are not dismissed immediately due to an initial lower fitness.

One fundamental consequence of using genetic algorithms is integrating into the optimisation process the concept that most of the core changes that defined life as we know it occurred at random. Moreover, many scientific discoveries are due to serendipity (reaching a goal, but not the one intended or not by the means that were thought to be the right ones) and many paths to certain inventions are made of steps whose objectives had nothing to do with the invention itself (*i.e.* vacuum tubes were invented in 1904 and made possible cathode ray tube monitors almost 30 years later). Had the invention been the objective in the first place, the goal may not have been reached .

An apparently logical way to solve a problem might turn out to be illogical or suboptimal. An example is choosing the distance to the exit as the fitness function to solve a maze. It is very likely to be stuck in a corner. It might seem obvious in that context, but choosing a fitness function then becomes difficult when dealing with biometrics. "Perturbing" your exploration of the solution space with the randomness and *novelty* brought by NeuroEvolution is akin to injecting the fitness function with the notion of local minima [116].

Since its conception, NEAT has been applied successfully to a multitude of tasks such as maze solving [117] and bipedal locomotion [118]; the latter shows how the evolutionary progress leads to a more "natural" walking cycle and it is closer to the human way of learning. Biometric modelling is different at the core as the task is not to solve *the* maze, but to solve **any** maze, as the target test example would obviously differ from the training one. NEAT has however been successful in more open-set tasks as automated videogame playing [119]. Although that work focused on rather primitive Atari games from the late 70s and early 80s where the degree of freedom in the input — and the number of possible outcomes — is rather limited, NEAT still managed to outperform other evolution-based systems. It also learnt higher-level gameplay concepts rather than simple input-output relations. In all aforementioned NEAT applications, augmented topologies always have the advantage of being considerably less complex than competing solutions, with the number of connections often being orders of magnitude less than solutions resulting from other deep learning approaches.

VI.2 Application to raw audio classification

NeuroEvolution is attracting significant attention [120]. Concurrently to the author's work on speaker verification and anti-spoofing (the latter being the topic of Chapter VII), he became aware of recent, successful attempts to utilize NEAT for audio-related tasks such as audio effect creation [121] and sound event detection [122]. In view of the computational demands for the training process, however, in the former work NEAT was applied using a variety of classic spectral/cepstral frame-based features instead of raw audio, while the latter used wavelet representations.

As explained in Section V.3.1 the use of some form of frame-blocked spectral or filterbank representation is common in ASV deep learning approaches, with all of the excep-

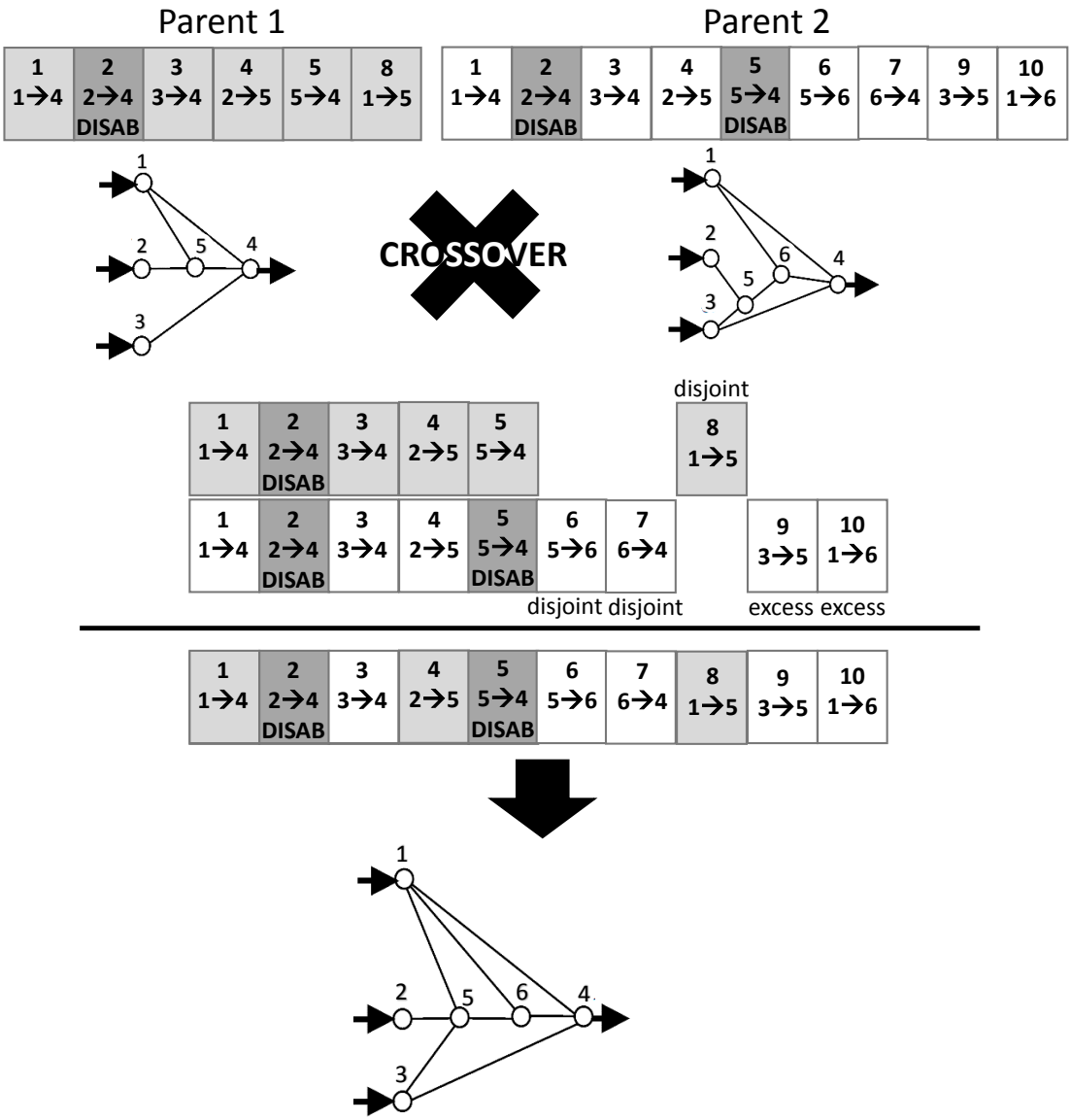


Fig. VI.4 *Gene alignment and generated offspring after crossover, reproduced from [114]: parent genes are aligned according to their innovation number, offspring genes are selected from either of the parents. More disjoint genes indicates less compatibility, excess genes (innovation numbers higher than the highest of the other parent) are usually kept, following the augmenting principle.*

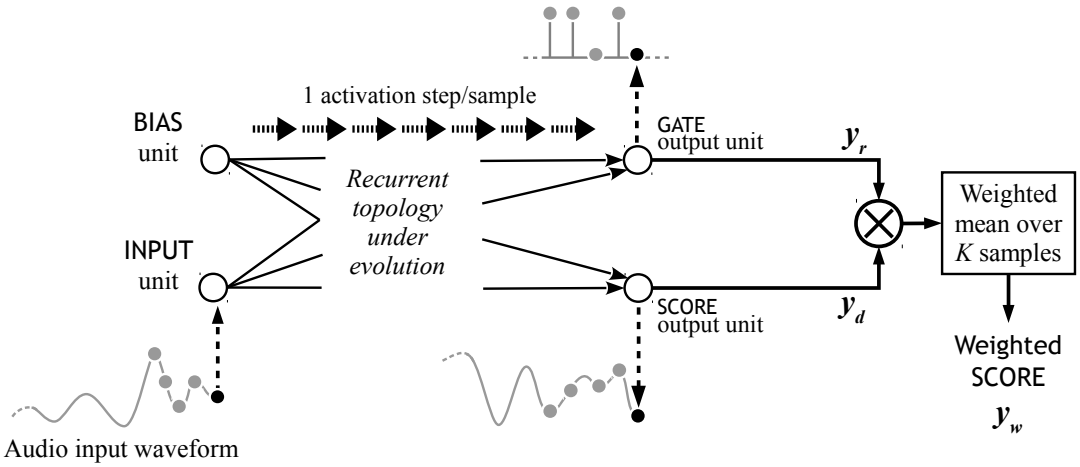


Fig. VI.5 *End-to-end setup and propagation scheme for audio classification. There is one activation step per input sample; the output rate is the same as that of the input.*

tions being very recent work [104–106]. Former NXP employee Daniel [123] reported the first application of NEAT to audio classification which operates directly on time-domain inputs.

In the approach proposed in [123], NEAT is applied with networks constrained to a specific input/output setup and propagation scheme. As illustrated in Fig. VI.5, inputs consist of one or more streams of raw audio. Each stream is mapped to an input unit and is propagated through the network sample-by-sample with one activation step for every sample. An additional bias unit is set and held to unity. Network outputs consist of one or more *score* units whose outputs y_d are multiplied by the output of a binary *gate* unit y_r. Except for the score and gate output units, which have identity and binary step activations respectively, all units have rectified linear activation functions. Connections can be made freely between *any* pair of units. As a result, evolved networks may contain cyclical unit connections (e.g. units connected to themselves or to other units which influence their input). This means that NEAT structures are inherently recurrent.

The rate of the output is identical to the sample rate of the input. In fact, the score output can be viewed as a new audio signal, the result of the network learning and applying to the input a transformation defined by the class to which the input belongs.

The gate will thus evolve to discard *output* scores which are deemed to be unreliable, so that the network places emphasis on samples that are most helpful to discriminate between different audio classes. Alternatively, the gate can be replaced by a *reliability output* yielding a non-negative, non-binary weighting factor y_r. The operation of the gate/reliability output is similar in principle to that of attention mechanisms [124] which have been applied previously to speech recognition [125]. For each time sample i, the weighted mean over K samples of the product of y_d and y_r yields the final weighted score y_w:

$$y_w[i] = \frac{\sum_{j=0}^{K-1} y_d[i-j] \times y_r[i-j]}{\sum_{j=0}^{K-1} y_r[i-j]} \tag{VI.1}$$

The behavior of each network is assessed according to a generic squared-error-based fitness function F:

$$F(y_w, g) = 1/\left[1 + \sum_{i=0}^{N-1} \left(g[i] - y_w[i]\right)^2\right] \tag{VI.2}$$

which reflects the distance between N weighted scores y_w and a ground truth signal g of classification labels, *e.g.* 0 or 1, making for a supervised approach.

VI.3 Truly end-to-end automatic speaker verification

With the potential for raw audio classification and evolved small-footprint topologies, this section reports the application of NEAT to conceive a truly end-to-end ASV system. As in the work of [123], all networks are constrained to share the common setup and propagation scheme illustrated in Fig. VI.5: there is one input stream, one bias, one output stream and a binary gate. The process described in Section VI.2 is applied to generate networks which distinguish between a given target speaker and a set of background speakers. Each iteration of the algorithm corresponds to one independent evolutionary process applied in speaker-dependent fashion. This process will produce a population of increasingly discriminative, speaker-dependent networks.

The evolutionary process is driven according to a new fitness function which is introduced below. Also described in this section is a mini-batch procedure which was found to be beneficial to the evolutionary process. Specific training and testing procedures are also presented.

VI.3.1 Fitness function

The fitness function in Eq. VI.2 does not necessarily reward separation between class distributions, but rather proximity to ground truth scores (*e.g.* 0 and 1). This behaviour becomes a problem when, after several generations, the two classes have only a minimal degree of overlap: a distance-based fitness function would reward a network that pushes the bulk of the distributions farther apart, without necessarily correcting previous classification errors (dashed-line PDFs in Fig. VI.6); conversely, a network which fully separates classes but which produces noisier or wider distributions (solid-line PDFs in Fig. VI.6) would be attributed less reward than another network which produces pure Gaussian, but slightly overlapped distributions.

An early search for an alternative, better suited to classification tasks such as ASV, investigated a fitness function based on the equal error rate (EER). The EER, though, only reflects the reliability of a classifier at a single operating point, *i.e.*, a fixed threshold. The area under the receiver operating characteristic curve (AUROC), in contrast, gives a measure of reliability which is independent from the operating point; it reflects the probability that the network will give a randomly chosen target sample a higher score than a randomly chosen impostor sample [126].

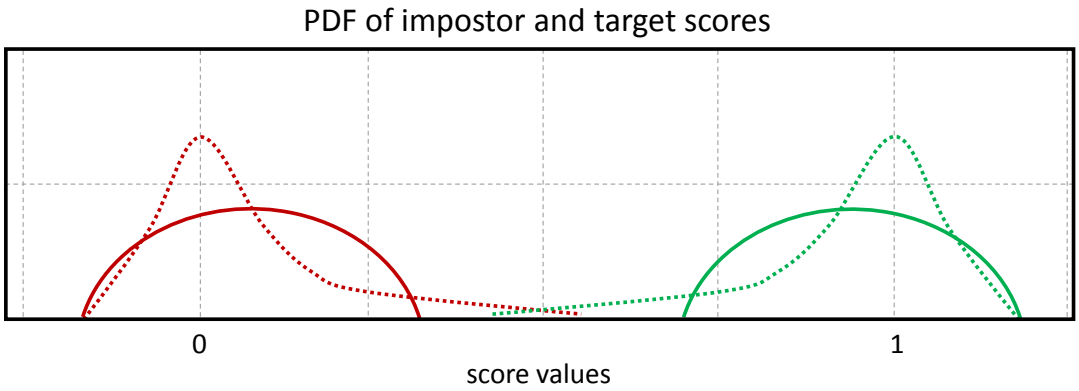


Fig. VI.6 *Example of two pairs of impostor and target scores distributions: by following the fitness function in Eq. VI.2, the solid-line distributions could have equal or even worse fitness than the dashed-line distributions, while it is clear that the latter would yield a better EER as they do not overlap. The shape of the curves does not stem from actual data distribution and is just for illustrative purposes.*

With notably better results, all work reported in this chapter was performed by replacing Eq. VI.2 with an AUROC function calculated using the *trapezoid rule* [127]. Although AUROC does not explicitly reward novelty [116], the concept is nevertheless present because of speciation (see section VI.1.2) and the use of mini-batching.

VI.3.2 Mini-batching

Inspired by a similar approach used in the *stochastic gradient descent* algorithm [128] to avoid over-fitting and convergence to local-optima, training is performed with a mini-batch process. The mini-batch process ensures that each generation of networks is trained using a different subset of data. This strategy promotes novelty during evolution since the training objective is shifted slightly upon every iteration. The same strategy also encourages generalization, namely networks which perform well across inter-session data. Finally, mini-batching also helps to reduce computational demands.

Each mini-batch consists of a fraction M_t of total target data and a fraction M_i of total impostor data. By way of example, with M_t=M_i=100%, every training iteration is performed using the *same* data; there is no mini-batching. With M_t=M_i=50%, training data is randomly shuffled and partitioned into two mini-batches. They are used in two subsequent iterations after which shuffling and partitioning is performed again before the next iterations of training.

VI.3.3 Training

The size of each population is fixed across generations and set to 150 networks. The algorithm is initialised (generation zero) with 150 minimal perceptron-like networks, all of which share the common setup described in Section VI.2. All input signals are normalised to within a range of [-1,1]. The choice of rectified linear unit activation functions results

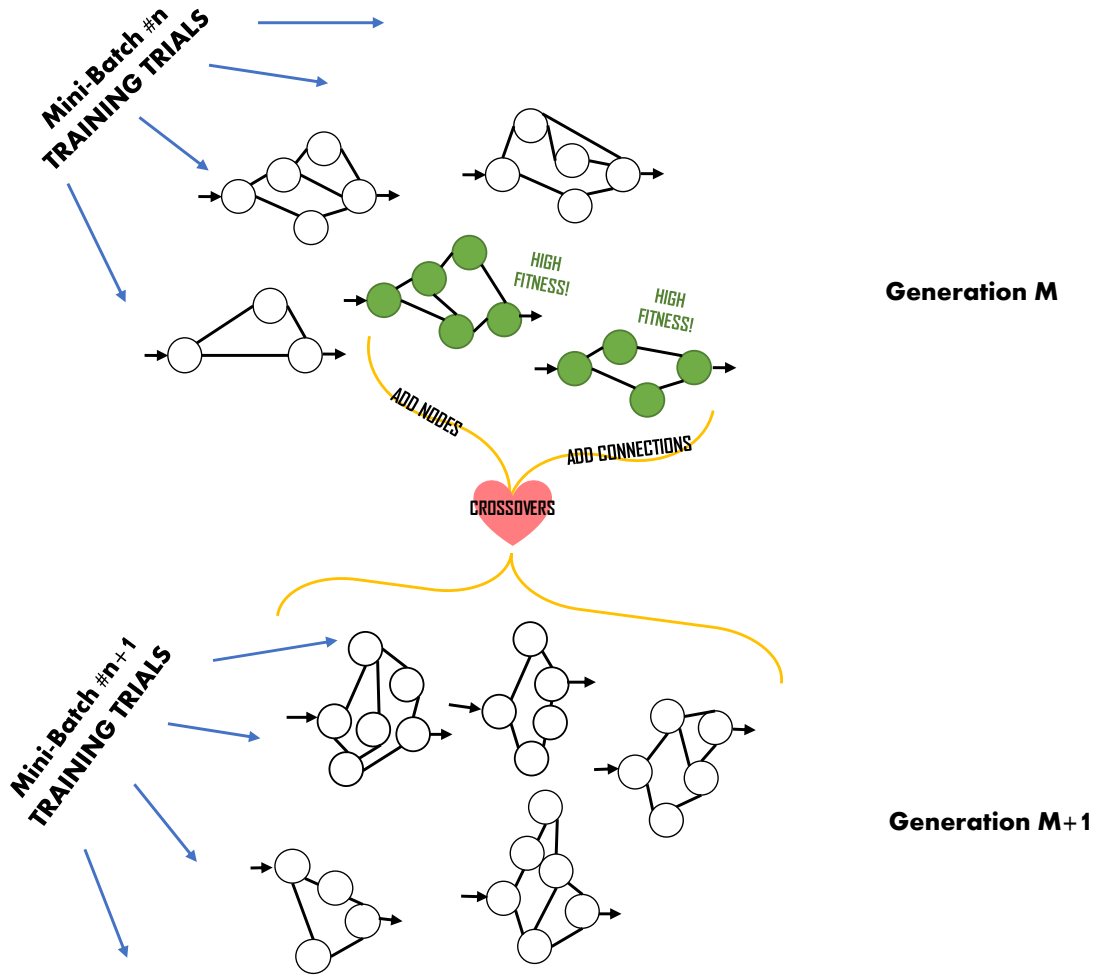


Fig. VI.7 *An illustration of 2 subsequent training iterations*

in faster processing (while giving similar performance), but also increases the chances of saturation. The random initialisation of weights within a $[-4, 4]$ range, combined with input normalisation was found to be effective in mitigating saturation. Every network in a given population is trained with the same mini-batch of data. Data containing either target or impostor speech is presented to each network in the form of overlapping segments of K samples. The system assigns to each segment a weighted mean score corresponding to $y_w[K-1]$ in Eq. VI.1. Networks are reset after the processing of each segment.

The fitness of each network is then determined according to the AUROC metric described in Section VI.3.1. The fittest networks of the population are then used to produce the next generation according to the procedure outlined in Section VI.1.2. The evolutionary process (illustrated in Fig. VI.7) is applied iteratively until the fitness converges.

Fig. VI.8 illustrates the evolution in fitness over 200 generations for an arbitrary target speaker. Each point on the graph corresponds to the population's fittest network for that generation. The solid blue profile illustrates evolution for the training procedure

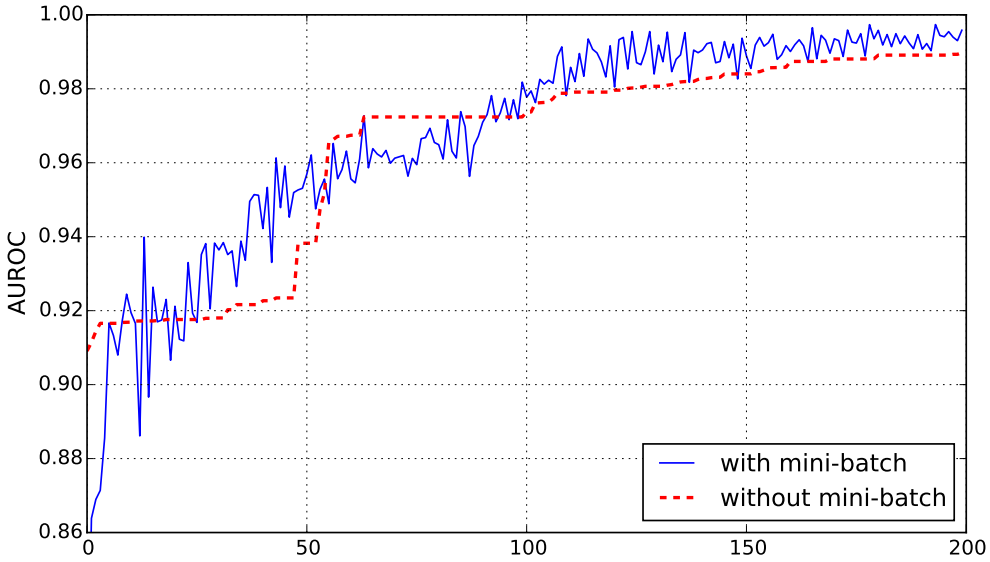


Fig. VI.8 *An illustration of evolution measured in terms of fitness (AUROC) for the fittest network of each generation. The solid blue profile illustrates the AUROC with mini-batch training whereas the monotonic dashed red profile shows the AUROC without mini-batch training.*

described above. Its non-monotonic nature is due to mini-batching; the data used at each iteration is different. The dashed red profile shows evolution with no mini-batching (M_t=M_i=100%); data used at each iteration is the same, hence the monotonic profile. While reducing processing time, mini-batching also results in faster learning.

VI.3.4 Network selection for evaluation

Once training is complete, it is necessary to select and evaluate the single best network. First, the 10 best networks of each generation are identified according to the AUROC fitness function. Second, the performance of each of the 10 best networks from each generation is reassessed using the *full* training set. Since it gives a more intuitive interpretation of performance in a practical application, selection is performed using the application-neutral EER metric. The network which produces the lowest EER among the 10 is designated as the *generation champion*. Finally, the generation champion associated with the lowest EER is designated as the *grand champion*, and selected for evaluation. Evaluation is performed using an independent test set.

VI.4 Experiments

This section describes experiments which aim to test the potential of the end-to-end ASV system described in Section VI.3. Text-independent experiments are performed on a non-standard corpus recorded internally at NXP, comprising of 10 male speakers. 3 baseline systems are proposed as a comparison: a traditional GMM-UBM system and two neural

network solutions.

The GMM-UBM system was used as a baseline in the earlier stages of this work; it was readily available from the experiments described in Chapters III and IV. The use a well-established Gaussian generative approach as a baseline, which was also familiar to the author, made it easier to design protocols for the experimental end-to-end system. Comparing the behaviour of the two systems guided the author on how to present the data in a way that allowed the E2E system to learn, avoid overfitting, and presenting an overall challenge.

The GMM-UBM system is the only baseline presented in the original publication of the work described in this chapter [8]. The two NN-based baseline systems previously unpublished and reported for the first time here, serve as a comparison to competing approaches belonging to the family of neural network and deep learning solutions, specifically those focused on small-footprint topologies.

VI.4.1 Baseline systems

The first baseline system is a standard 64-component GMM-UBM system [20]. Features are standard 19th order MFCCs. These are appended with delta and double-delta parameters thereby giving features of 57 coefficients.

Speaker models are derived from the maximum a posteriori adaptation of the UBM (see Section II.3.2). Scores are log-likelihood ratios given the speaker model and the UBM.

The neural network solutions consist of (i) a CNN system based on the 'cnn-trad-fpool3' model in [129] and (ii) a DNN based on the *rank-constrained* (RC) topology network in [130]. The author acknowledges that the NN-based baseline systems are not state-of-the-art implementations of deep learning ASV approaches, but are to be intended as comparisons closer in terms of technology to the end-to-end approach than the conventional GMM system. The NN approaches chosen for this work also place a strong focus on low-resource, small-footprint network architectures.

In [129] and [130] these architectures are employed for keyword spotting, which is a multi-class word recognition task. They are here adapted to speaker verification by reducing the possible output classes to encompass the target speaker and the background/garbage model only. Both publications report models to have in the order of hundreds of thousands of connections (244k and 102k respectively). The RC-DNN topology was developed with limited inference resources in mind; computational effort at test time is approximatively 0,001% of the CNN system. The principle behind the parameter reduction resides in approximating the weight matrix of the 2-dimensional input with a product of two separate weight vectors for time and frequency.

Input for each network typology is 40-dimensional MFCCs or 40 log-mel filterbank energy features, totalling 4 different NN-based configurations. The temporal dimension is fixed to 1 second (100 frames) for both training and testing trials. The actual number of parameters for the proposed implementation is 940k for the CNN and 750k for the RC-DNN.

Table VI.1: *Results for the baseline systems and end-to-end system in terms of EER for the training and test set for the two target speakers.*

	Speaker #1		**Speaker #2**	
	Training	Test	Training	Test
GMM-UBM	0%	9.5%	0%	6.9%
CNN$_{MFCC}$	0.8%	6.8%	0%	2.5%
CNN$_{logmel}$	0.4%	5.8%	0%	1.0%
RC-DNN$_{MFCC}$	0.6%	9.0%	0.2%	1.7%
RC-DNN$_{logmel}$	0.8%	6.5%	0.2%	1.2%
End-to-end	0.8%	5.3%	1.0%	9.4%

VI.4.2 NXP database and experimental protocols

Experimentation with standard NIST Speaker Recognition Evaluation datasets [131], RSR [37] or RedDots [46] are currently impracticable on account of the prohibitive training time given the current implementation of the end-to-end system. Being consistent with the objective to evaluate the potential of the algorithm, this section reports a set of proof-of-concept experiments using a non-standard, proprietary database of speech signals collected from 10 male speakers. Text content consists of 10 of the 30 Harvard sentences which comprise the TIMIT database [132]. Each speaker provides approximately 5-6 minutes of speech which is recorded in 9 sessions over the course of one month. Recordings were collected in a quiet office with a laptop at a sampling rate of 16 kHz and 16-bit precision. Utterances were normalized by the active speech level estimated according to the ITU-T P.56 standard [133].

Among the 10 speakers, 2 are enrolled as targets. The training set consists of 6 of the 10 sentences uttered by the target speaker and the first 5 impostors. The test set consists of the other 4 sentences uttered by the target speaker and the remaining 3 impostors, thus achieving considerable phonetic separation between sets. The split between training and testing sentences is the same for both target speakers. Total target training data amounts to approximately 3.5 minutes of speech per speaker. Total impostor training data is in the order of 14 minutes duration.

For the end-to-end system, target data is partitioned into two mini-batches (M_t=50%). Since impostor data is more plentiful, it is partitioned into five mini-batches (M_i=20%) and used as background data for the baseline system. The average duration of training utterances is 3.25 seconds. For the assessment and testing of all systems, one trial corresponds to one entire recording. Accordingly, K (see Section VI.3.3) is set to $3.25 \times 16000 = 52000$ samples for training, and to the number of samples in each trial for testing. Audio files used by the baseline systems are preprocessed with silence removal. This step is not performed for the end-to-end system.

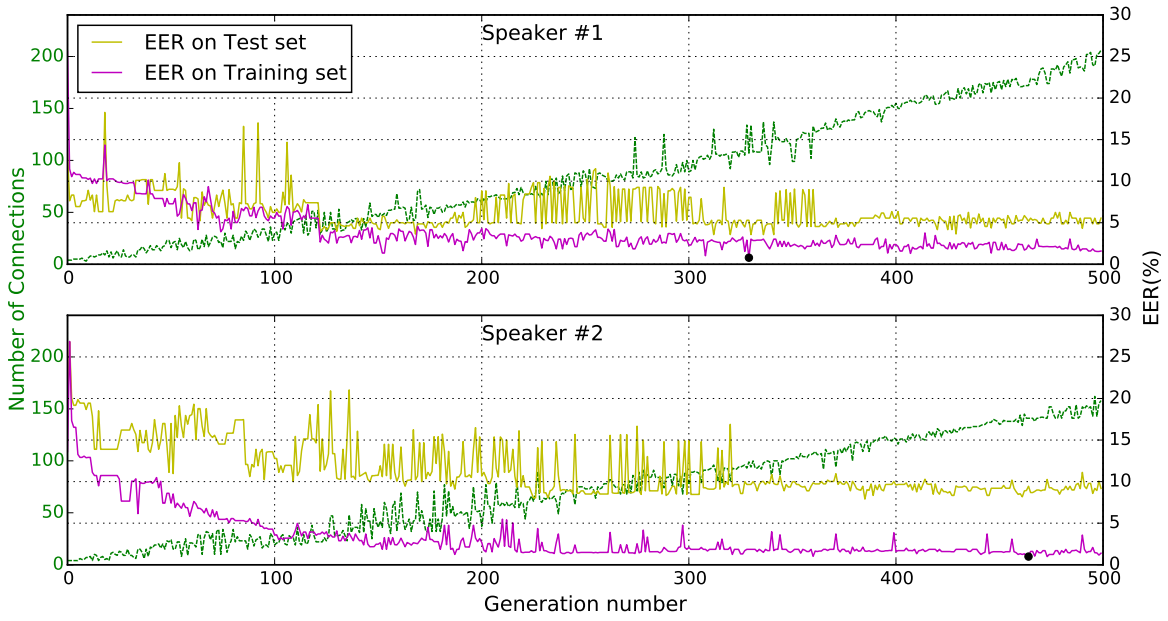


Fig. VI.9 *Number of connections (green dashed profiles) and equal error rate (EER) of the first 500 generation champions for target speakers 1 (top) and 2 (bottom). EER profiles are shown for training data (magenta/dark) and testing data (yellow/light profiles). Black dots signify the grand champion, chosen according to the lowest-EER on the full training set.*

VI.4.3 End-to-end system: augmentation and generalisation

The evolutionary process and network evaluation and selection procedure described in Section VI.3.4 are depicted in Fig. VI.9. Two processes are performed and illustrated independently for the two target speakers and for 500 generations. The solid magenta (dark) profile in each plot shows the EER obtained by each generation champion assessed using the training data. EER profiles exhibit the expected evolution trend, namely a steady decrease from above 30% to less than 5% within 150 generations. The lowest EERs obtained by grand champion networks are 0.8% for speaker 1 (generation 329) and 1.0% for speaker 2 (generation 464) marked by black dots. Solid yellow (light) profiles show EERs for generation champions assessed on test data. As expected, performance on independent data is worse. Nonetheless, the selected grand champions are among the best performing networks when assessed using test data.

A summary of performance for baseline systems and the end-to-end system is presented in Table VI.1. For the latter, results for both train and test datasets concern the grand champion network selected for each speaker. For the test set, grand champions yield EERs of 5.3% and 9.4%, whereas the GMM system delivers EERs of 9.5% and 6.9%. As expected, CNN models perform overall better compared to less complex RC-DNNs, with the exception of speaker #2 for the MFCC configurations. Feature-wise, log-mel energies seem to bring consistently better performances to both types of architectures. The end-

to-end system does not follow the general trend of lower EERs for speaker #2 versus speaker #1: while on training data the EER is kept below 1%, on test data speaker #2 proves to be more difficult, as is clearly depicted by the higher yellow profile on Fig VI.9. This situation is almost mirrored with the GMM system, suggesting that very different cues are learnt while operating in the time domain. The end-to-end system, albeit in an early stage of development, delivers the lowest EER for speaker #1 compared to all other systems.

The upper green dashed profiles in Fig. VI.9 show the number of connections of each generation champion. As evolution proceeds, networks are steadily augmented with new nodes and connections. In general, network augmentations cause decreases in EERs for the training set, with 112 and 138 connections for speaker 1 and 2 grand champions, respectively. These networks are orders of magnitude less complex than usual, deep layered structures (*c.f.* $\sim 200k$ connections for the most compact model reported in [107]). Networks with such a reduced parameter space are inherently less prone to over-fitting since they do not have the capacity to learn a direct input-output correspondence.

The gates of the grand champion networks prune an average of 46% of output data (in both speech and non-speech intervals) — the average for speaker 1 is 40% whereas that for speaker 2 is 53%. This percentage is consistently higher than for the baseline systems for which silence removal prunes an average of 35% of data (obtained by calculating the speech active level according to ITU-T P.56[1] thresholding at 15.9 dB). The behaviour of the gate was observed for a small number of trials, and showed a periodic opening and closing as opposed to an energy- or amplitude-related activation. An illustration of the gate behaviour is shown in Fig. VI.10: as a purely empiric observation, it seems the gate learnt to open on high output samples for targets and low output samples for impostors, effectively rendering the output signal less periodic and more score-alike. The active speech level used in the baseline systems usually results in cutting the low-amplitude parts preceding and following the utterance. These findings show that the GMM and end-to-end systems exploit data in a different way.

VI.5 Further experiments: End-to-end system on NIST SRE16 data

The work published in [8] demonstrated a successful proof of concept with a small, non-standard database. The work reported in this section aims to test the proposed end-to-end system with a publicly available corpus, namely the NIST SRE dataset, whose various "editions" during the years (see Section II.5.1) serve as standard databases within the ASV research community.

All experiments reported here were performed on a subset of the NIST SRE16 development set (see Section II.5.1.d). At the time when the experiments were carried out, the SRE16 corpus was the most recent NIST dataset for which results had been published. Being a very recent corpus, collected explicitly for the most important speaker recognition evaluation poses a more difficult challenge to the proposed system compared to the NXP database.

[1] http://www.itu.int/rec/T-REC-P.56-201112-I/en

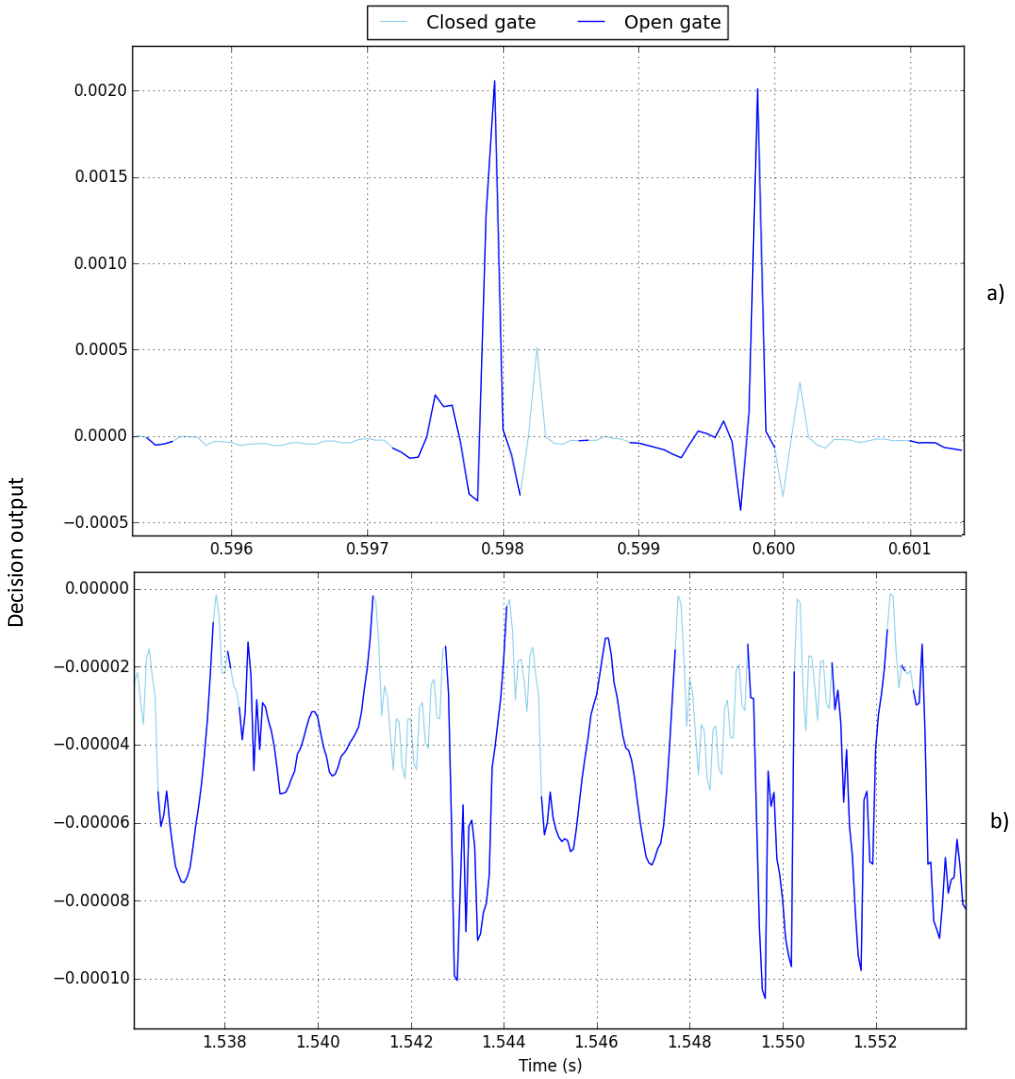


Fig. VI.10 *The gate behaviour on a target (a) and an impostor (b) trial: darker samples signify an open gate. Note how it seem to learn when to open in order to give higher averaged scores to target trials and lower scores to impostor trials, effectively "helping" the decision output.*

As illustrated in Table VI.2, the labelled data of the development set (see Section II.5.1.d for a description of the NIST SRE16 corpus) is comprised of data belonging to 20 speakers, which according to the official protocols is used to enrol 80 different models. Even with such a modest number of speakers, a full-protocol experiment is computationally prohibitive (see Section VI.4.2), therefore a subset of the data for 7 models belonging to 7 of the 10 male speakers is selected. In order to have at least one minute of training data per speaker, the models were selected among those enrolled with 3 telephone conversations. The background data, complying with SRE16 guidelines[2], consists of SRE04, 05, 06, Switchboard and Fisher corpora datasets, comprising 3420 recordings of male speakers. For the end-to-end system, the background protocol is also subsampled by a factor of 10 to speed up the training and network selection process, as the algorithm still requires to test on the full training set (which includes background data) to select generation champion networks (see Section VI.3.4).

The setup for the end-to-end system is left unchanged from Section VI.3, except for a few adjustments:

- The training segment length K is set to 3 seconds, equal to $3 \times 8000 = 24000$ samples.
- Since NIST SRE16 data involves one side of a telephone conversation, this results in the presence of long periods of silence in each file. In order to avoid completely silent segments, files are preprocessed with silence removal which leaves on average 2 minutes of training data per speaker and 10 hours 48 minutes of background data. In order to approximately recreate the same silence-speech balance of the NXP database, only silence periods longer than 500 milliseconds were cut.
- Mini-batch parameters are set to $M_t = 66\%$ and $M_i = 0.3\%$

While the purpose of this experiments is to test the end-to-end system with data from a standard corpus, the results are compared with the baseline system reported in [10] as "ICMC IV PLDA" (henceforth referred as ICMC) developed at EURECOM. This was done for two reasons: (i) individual scores for the system were made available to the author, allowing a 1:1 comparison of the two systems with any subset of trials and (ii) the ICMC system is a developed and optional sub-system which was used in the I4U consortium submission to the NIST SRE16 evaluation.

The ICMC system is of the i-Vector PLDA type (see Section II.3.4) and uses constant Q Mel-scaled cepstral coefficient (ICMC) [58] features. This system also utilises additional SRE08 data to train the PLDA hyperparameters; see [10] for further detail. Reported in Table VI.3 are the results for the male speakers of the development set. Comparative results in Table VI.4 are reported in the form of EER per speaker. There is no global- or language-dependent EER because, as per with the NXP database experiments, each speaker-specific network has very different score dynamics and calibration or score normalisation methods for the end-to-end system have not been developed yet.

Individual scores for the ICMC system were made available to the author. This allows to calculate the global EER for the whole SRE16 development set, group it by gender and/or language, as well as by speaker. Comparisons corresponding to each model are actually possible, in addition to comparing the ICMC speaker-dependent EERs with their respective language-dependent EERs reported in Table VI.3.

Table VI.2: *Statistics for the NIST SRE16 development labelled data*

# Speakers	20
# Models	80
# Calls	200
# Target trials	4828
# Impostor trials	19312
Languages	Mandarin, Cebuano

Table VI.3: *ICMC IV PLDA results for **all** the male speakers and models of the NIST SRE16 development set, grouped by language. This table shows how each of the 7 models in TableVI.4 is representative of the whole male set for the corresponding language.*

	Mandarin	**Cebuano**	**Averaged**
EER (%)	8.4%	32.2%	20.3%

Table VI.4: *Results on NIST SRE16 development set data for the end-to-end system and the ICMC IV PLDA system reported in [10]. Results between parentheses are for networks chosen a posteriori (not following the grand champion selection policy) and are just reported for observations*

Model ID	**Language**	**E2E**	**ICMC**
1008	Mandarin	46.5 (29.6)%	6.7%
1011	Mandarin	29.4 (15.2)%	0.3%
1036	Mandarin	23.0 (15.8)%	9.4%
1050	Mandarin	24.6 (18.0)%	8.1%
1039	Cebuano	16.4 (14.1)%	12.9%
1043	Cebuano	25.3 (**15.4**)%	16.9%
1078	Cebuano	**21.6** (**20.0**)%	36.2%
Average 1	Mandarin	30.9 (19.7)%	6.1%
Average 2	Cebuano	**21.1** (**16.5**)%	22.0%

Results for the E2E system are reported in 2 forms: the first is the EER for the grand champion, selected following the procedure explained in section VI.3.4; the second, illustrated in parentheses in Table VI.4, is the EER obtained by selecting the best generation champion on the **test** set, therefore this can only be done a posteriori. Neither network is trained with test data, but the way the latter is chosen is rightfully considered *cheating*. Reporting this EER, though, brings some noteworthy information: in the pool of generation champions (which is already a selection based on training data) there are some networks that perform considerably better than grand champions, even if they are not chosen as grand champions.

On the one hand this is encouraging, because it shows the learning power of E2E networks (results are closer to or better than the ICMC system, especially on the Cebuano language), but it also proves that the grand champion selection policy is less than optimal for databases such as the NIST SRE16 data which exhibit a language mismatch with the background data (all background data is in English) but not with impostors at testing (the used protocol only involves language-matched trials).

With all of this considered, even by ignoring the chosen-a-posteriori champions, in one instance the E2E approach outperforms the ICMC system, whose weakness seems to be Cebuano speakers which are, conversely, the easier ones for the end-to-end system. This once again reinforces the thesis that the E2E system is capable of learning information that is complementary to that learnt by other approaches.

VI.6 Conclusions

Deep Neural Networks and deep learning are today at the state of the art in Automatic Speaker Verification. The literature shows a trend to replace hand-crafted features with alternatives learnt automatically with machine learning. Furthermore, the complexity of deep network architectures, besides the associated computational efforts, makes explaining or interpreting what makes them work considerably challenging. Jointly-optimised front-end and back-end approaches, referred to as end-to-end approaches, rarely use raw audio data as input and exhibit fixed or very constrained architectures.

This chapter reports an end-to-end approach to automatic speaker verification (ASV) based on the NeuroEvolution of augmenting topologies (NEAT) algorithm. In contrast to the existing state of the art, the proposed algorithm avoids the use of hand-crafted features by processing raw audio and optimizes network weights and topologies in an entirely end-to-end fashion. Less complex topologies with a low memory footprint are well suited to embedded implementations.

The first set of experiments pertaining to the relatively small proprietary NXP database, compares the E2E system with GMM- and NN-based systems. Results show that the end-to-end system is at least competitive. A second set of experiments performed on a subset of the standard NIST SRE16 corpus, a much more challenging task involving cellphone-quality speech in non-English languages, while all the background data is in English. Results on this dataset yield considerably worse results, but some a-posteriori observations show that some networks in the population achieved better performances than the chosen grand-champions, putting the "blame" on the selection procedure, which

[2] https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016

needs further refinement. Nevertheless, on Cebuano speakers, the end-to-end approach obtained results comparable to the ICMC system.

A particularly appealing feature of the end-to-end approach is the gate, which acts as a form of built-in attention mechanism which serves to distinguish the most reliable information in the network output. This aspect of the end-to-end solution requires further investigation in order to interpret its behaviour with respect to information present in the acoustic input.

These findings suggest that the end-to-end approach merits further attention. Experimentation with unabridged, official protocols have to become feasible. This will require improvement to computational efficiency; the current CPU-only implementation makes larger-scale experimentation impracticable. Future work should investigate efficient implementations which exploit hardware acceleration, non-binary gates for soft, rather than hard weighting of output score samples, experimentation with longer duration training and testing, and a scoring method taking into account the order of the outputs rather than the plain average over a number of output samples.

The application of augmenting topologies to raw audio is still in its infancy, and with the challenge being great and variate, the approach is understandably not up to the performance of state-of-the-art systems. Nevertheless the room for improvement is equally large in many aspects. This work may well bring improvements in end-to-end system performance and/or expose application settings for which the proposed approach may excel.

Chapter VII

Augmenting topologies applied to anti-spoofing

This chapter describes the adaptation of the NEAT approach to *spoofing* detection. Despite the progressive adoption of automatic speaker verification as a reliable authentication method, vulnerabilities to spoofing (also known as *presentation attacks*) give reason for caution. Without adequate countermeasures, fraudsters can manipulate the normal operation of an authentication system by masquerading as genuine users and hence gain unauthorised access to protected resources or services. Vulnerabilities to presentation attacks are clearly inadmissible. In addition to the immediate security concerns, they undermine confidence in ASV technology.

With the end-to-end system described in Chapter VI, each end-to-end network is a discriminative, binary classifier, learnt according to its own independent evolution process. One of the issues in applying the proposed end-to-end approach to ASV in its current CPU-only implementation relates to its extensive training time, which prohibits large-scale experimentation involving a significant number of speakers (see Sections VI.4 and VI.5). With anti-spoofing however, the task is to discriminate between genuine and spoofed speech, *i.e.* only a single model. Accordingly, NEAT can be applied to anti-spoofing to recordings collected from multiple speakers by using just one classifier, allowing for faster experimentation and optimisation with the current end-to-end pipeline.

This is not to say anti-spoofing is independent from ASV. On the contrary, it should be complementary and seamlessly integrated in any ASV system, though it can be and it is often treated as a separate task. Indeed, anti-spoofing is a relatively new field compared to ASV and related research has advanced considerably in the last three years. Anti-spoofing is a particularly difficult pattern classification problem since the characteristics of spoofed speech vary considerably and can never be predicted with any certainty in the wild. The design of features suited to the detection of unpredictable spoofing attacks is thus a staple of current research. End-to-end approaches to spoofing detection which exploit automatic feature learning have shown success and offer obvious appeal. Chapter VI saw the application of augmenting topologies to raw-audio ASV, the objective of this chapter is to investigate if the discriminative nature of the proposed end-to-end approach has potential as a spoofed speech detector.

To adapt the end-to-end ASV system described in Chapter VI to spoofing detection,

a new fitness function is introduced. While not explicitly developed for anti-spoofing, the new fitness function has proven successful for the first time when applied to the task. This fitness function is designed to exploit the temporal aspect of evolution: it stores information about data that was correctly classified during past generations in order to reward true progress instead of mere performance.

The speaker-independent approach to spoofing detection made it feasible to test the end-to-end system using the official protocols and baseline system of the ASVspoof 2017 database of *bona fide* speech and *replay* spoofing attacks, which involve the (surreptitious) capture and subsequent playback to the ASV system of a speech sample captured from a genuine speaker/user.

VII.1 A brief overview of anti-spoofing

It is known that ASV systems can be vulnerable to spoofing attacks in the form of *impersonation*, *synthetic speech*, *converted voice* and *replay* [134]. Impersonation (the imitation of a target speaker by another person) requires a certain skill and is generally considered to pose only a modest risk [135]. While the threats posed by synthetic speech and converted voice are potentially severe, given that their implementation requires specialist expertise that only a few have, the actual risk may be relatively low. Replay attacks arguably present the greatest threat as they can be mounted easily with widely available, consumer-grade audio recording and playback devices (e.g. smart phones) and can be especially difficult to distinguish from genuine, bona fide speech samples. The detection of replay attacks is the focus of this chapter

Efforts to develop spoofing countermeasures, also known as presentation attack detection (PAD) systems, are now well under way; the study of spoofing countermeasures for ASV is today an established area of research [136]. The first competitive evaluation, namely the ASV spoofing and countermeasures (ASVspoof) challenge [137], was held in 2015. It promoted the development of countermeasures to protect ASV from voice conversion and speech synthesis attacks. The second edition of ASVspoof, held in 2017, switched focus to the mitigation of replay attacks [138–140].

The many submissions to the ASVspoof 2017 challenge can be classified into one of two different approaches. The first set of approaches involves the combination of hand-crafted features with generative classifiers such as Gaussian mixture models (GMM) and i-vectors/PLDA systems, e.g. [141–148]. The second of approaches explored the use of discriminative classifiers such as support vector machines (SVMs) and deep neural networks (DNNs) [113, 143, 146–150].

Deep learning techniques, in particular, proved to be especially successful, with five of the top ten performing systems submitted to the ASVspoof 2017 challenge employing some form of automatic feature learning and/or classification[1]. The work in [146] used convolutional neural networks for the automatic learning of features from magnitude spectrograms with a combination of convolutional and recurrent layers in an end-to-end solution. This success serves as a motivation to further explore the potential of deep learning, especially towards end-to-end solutions.

[1] A summary of top submissions is available at http://www.asvspoof.org/slides_ASVspoof2017_Interspeech.pdf

End-to-end approaches to anti-spoofing have obvious appeal. As explained in sections V.2.1 and V.3.1, automatic approaches to feature learning are bringing advances in performance and the quest for better-performing, hand-crafted features is a staple of current research. Research in anti-spoofing is comparatively embryonic, as the history of benchmarking evaluations spans less than four years. The natural variation in spoofing attacks is so great that makes hand-crafted feature design especially difficult. In the absence of an extensive body of background knowledge or proven features, and with the availability of large datasets of spoofed speech, automatic feature learning and end-to-end solutions present an opportunity to fast track progress.

VII.2 NEAT setup

The raw audio setup described in Sections VI.2 and VI.3.3 is applied for this task, with one notable exception: the author's work on ASV, as well as that in [123] adopted a flexible streaming/segmental or file-based approach, where the segment length K is adjusted accordingly (see Eq. VI.1). Conversely, all work on anti-spoofing relates solely to file-based processing where K is always set to the number of samples in a file. Given the focus of this section upon anti-spoofing rather than automatic speaker verification, mini-batch notations M_t (target speaker) and M_i (impostor speakers) are replaced with M_b (bona fide speech) and M_s (spoofed speech). This notation is adopted throughout the remainder of this section. Other parameters are left unchanged, with the main, novel contribution of this work being the new fitness function. This is explained below.

VII.2.1 Ease of classification

Experiments to evaluate the performance of NEAT for anti-spoofing using the previously reported fitness functions [8, 123] showed a tendency to oscillate around local optima, namely networks in subsequent generations that correct previous classification errors while introducing new ones. Such oscillations can be avoided by using an enhanced fitness function which rewards *progress* rather than raw performance. Progress infers the learning of better networks which correct previous classification errors *without* introducing new ones.

An expression for fitness which rewards progress requires the definition of a measure of segment classification ease. Given the file-based approach adopted for the work in this chapter, a segment always corresponds to one trial or file. Intuitively, this is proportional to how high or how low is the score for segment s compared to the average impostor (spoofed) or target (bona fide) scores respectively; For every network n and **bona fide** segment s with score θ_s, the classification ease is given by:

$$l_{s,n} \leftarrow 1 - \frac{\#\{\textbf{spoofed} \text{ segments with score} > \theta_s\}}{\#\{\textbf{spoofed} \text{ segments}\}} \tag{VII.1}$$

where the right-most term is akin to the false acceptance rate for the given threshold. Conversely, for every **spoofed** segment with score θ_s, the classification ease is given by:

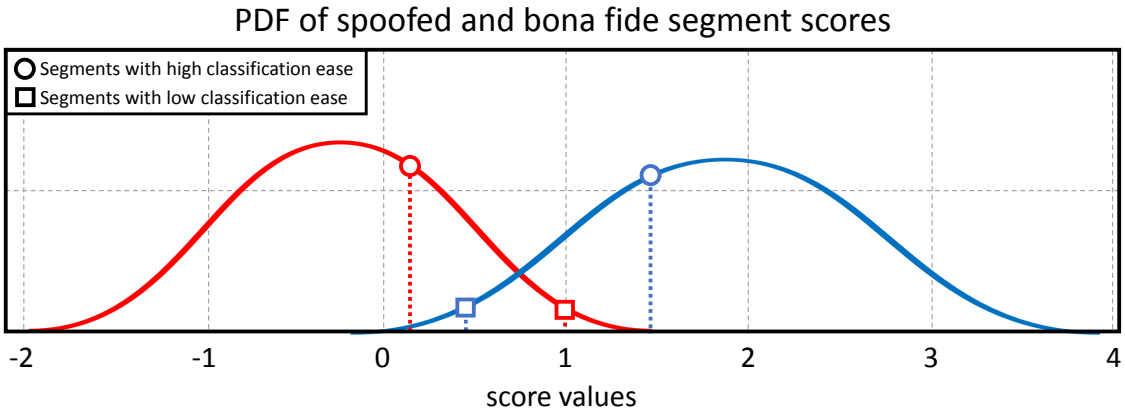


Fig. VII.1 *The position of segments with high or low classification ease depends on how many segments of the other class were given higher or lower scores for bona fide or spoofed segments, respectively.*

$$l_{s,n} \leftarrow 1 - \frac{\#\{\textbf{bona fide} \text{ segments with score} < \theta_s\}}{\#\{\textbf{bona fide} \text{ segments}\}} \tag{VII.2}$$

where the right-most term is now akin to the false rejection rate for the given threshold. An illustration of a few examples of segments with relatively low and high classification ease is depicted in Fig VII.1.

A *pooled* measure of the classification ease may then be obtained by averaging the classification ease over the number G of networks in the population:

$$p_s \leftarrow \frac{\sum_n l_{s,n}}{G} \tag{VII.3}$$

where $l_{s,n}$ is set according to Eqs. VII.1 or VII.2 depending on whether segment s corresponds to a bona fide or spoofed trial respectively. A measure of network fitness F is then estimated across all segments according to:

$$F = \frac{\sum_s l_{s,n}(1 - p_s)}{\sum_s (1 - p_s)} \tag{VII.4}$$

where $(1 - p_s)$ acts to weight the contribution of the classification ease for segment s, and network n.

This approach to fitness estimation is from here on referred to as the ease of classification (EOC). The EOC fitness function was developed in collaboration with Adrien Daniel while he was employed at NXP Semiconductors.

According to Eq. VII.4, the correct classification of segments that were already correctly classified by networks in an earlier generation thus contributes little to the estimation of fitness for networks in the subsequent generation; there is little reward for learning what was already known by an earlier generation. The EOC approach to fitness estimation steers evolution to classify correctly a diminishing number of *difficult* segments. It is

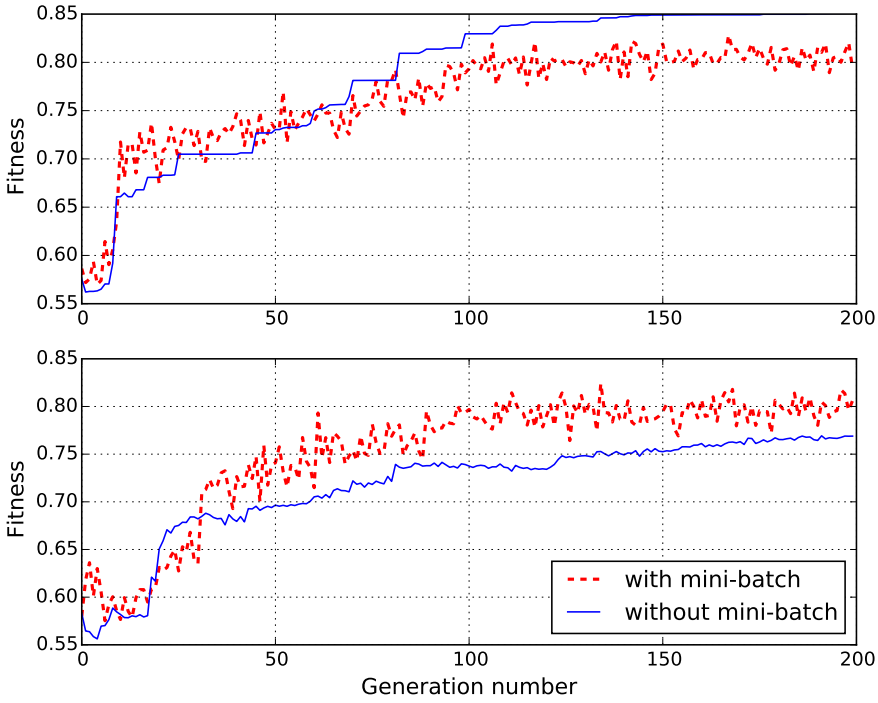


Fig. VII.2 *An illustration of fitness evolution for the fittest network of each generation when using AUROC (top) and EOC (bottom) fitness functions on ASVspoof 2017 data. The dashed red profile illustrates the fitness evolution with mini-batch training whereas the mostly-monotonic blue profiles show the fitness without mini-batch training.*

further stressed that, for the work described in this chapter, a segment always corresponds to a file/trial.

VII.2.2 Training

Experiments were conducted with both AUROC (Section VI.2) and EOC (Section VII.2.1) fitness functions, with and without mini-batching. Audio signals containing either bona fide or spoofed speech are fed to each network segment-by-segment (file-by-file) and the network is trained in order to distinguish between the two. The fitness function is evaluated with K in Eq. VI.1 set to the number of samples in each file. All networks are reset/flushed after the processing of each file.

At each iteration (generation), a subset of the fittest (best performing) networks among each species is determined and used to evolve the next generation of networks according to the procedure outlined in Section VI.1.2. Evolution proceeds either until the fitness converges or until a pre-determined maximum number of generations is reached.

Fig. VII.2 depicts the improvement in network fitness (vertical axis) over 200 generations (horizontal axis). Illustrated is the evolution in fitness for four different configurations: the two fitness metrics AUROC and EOC, both with and without mini-batch training ($M_b = 25\%$ and $M_s = 33\%$). Each point on each profile shows the fitness of the

single, fittest network among the 150 in the population. In both graphs the dashed red curves relate to mini-batch training. Neither profile is monotonic since the data changes at each generation. Conversely, solid blue curves show fitness without mini-batch training ($M_b = M_s = 100\%$), hence the largely monotonic profiles (the fitness of EOC optimised networks is not strictly monotonic on account of the different weights applied to each segment during fitness estimation, as described in Section VII.2.1).

Profiles in Fig. VII.2 show that mini-batching is of more benefit when used with the EOC fitness function. Changes in training data can be interpreted as optimisation towards a moving target. This fuels novelty instead of over-fitting to a fixed training set. These observations would suggest a potential for better generalised spoofing detection. It should be noted, however, that the final objective is not higher fitness for training data, but the *classification reliability* assessed using test data.

VII.2.3 Testing

Networks with high measures of fitness may not necessarily be those which give the best performance in terms of the spoofing detection EER. This is especially true when using mini-batch since one random subset of training data could be fortuitously easier than another subset (or indeed the full set). In addition, measures of fitness derived using the EOC fitness function may not be especially well correlated with classification performance; increases in EOC reflect the learning of new information rather than raw performance. Moreover, while the AUROC and EER can both be obtained for any given set of trials, EOC cannot be correctly calculated at test time because there is no *pooled classification ease* (see Eq. VII.3), since there is no previous classification history regarding test trial files.

These reasons make even more crucial the selection procedure of generation champions and grand champion described in section VI.3.4, which is here applied following the same steps. The grand champion network selected for testing/evaluation is then used without further modification.

VII.3 Experimental setup

This section describes the database, protocol and metric used for all experiments reported in the remainder of this chapter. Also described is the baseline system and specific configuration details for the end-to-end approach to anti-spoofing.

VII.3.1 Database, protocol and metric

Experiments were performed using Version 2.0[2] of the ASVspoof 2017 database [151]. The database originates from the RedDots database[3] which was collected by volunteers from across the globe using mobile devices, in the form of smartphones and tablet computers. While the RedDots database was collected to support research in text-dependent automatic speaker verification, the ASVspoof 2017 database was adapted from it in order to support research in anti-spoofing. It contains sets of bona fide (genuine) and replayed

[2] http://dx.doi.org/10.7488/ds/2301
[3] https://sites.google.com/site/thereddotsproject/

Table VII.1: Statistics of the ASVspoof 2017 database version 2.

Subset	# spk	# replay sessions	# replay configs	# utterances bona fide	# utterances replay
Training	10	6	3	1507	1507
Devel.	8	10	10	760	950
Eval.	24	161	57	1298	12008
Total	42	177	61	3566	14466

speech [139, 152, 153]. In order to simulate replay spoofing attacks, the bona fide partition of the ASVspoof 2017 database was replayed and then recaptured using a variety of different loudspeakers and recording devices in heterogeneous acoustic environments.

The standard protocol relates to a partition of the database into training, development and evaluation subsets, details of which are presented in Table VII.1. The three subsets are mutually disjoint in terms of speakers and of data collection sites. Experiments reported in Section VII.4 were performed with the extended protocol whereby both training and development were performed with pooled training and development partitions (train+dev). The evaluation subset contains data collected using 57 replay configurations, 49 of which differ to those used in the collection of the training and development subsets. Differences in replay detection performance between the training/development and evaluation subsets serve to gauge the generalisation of spoofing countermeasure solutions.

The ASVspoof 2017 evaluations assessed the performance of spoofing countermeasures in isolation to automatic speaker verification. All results are reported in the form of the EER%.

VII.3.2 Baseline systems

The ASVspoof 2017 Version 2.0 database was released in order to correct data anomalies detected subsequent to the official evaluation. This new version of the corpus was first presented in [154] along with a new baseline system, here referred as baseline 2.0. The original ASVspoof baseline for Version 1.0 is included, since at the time when the work reported in this chapter was carried out and published in [9], it was the only system for which results relating to database Version 2.0 were available [4].

Baseline Version 1.0 uses a constant Q cepstral coefficient (CQCC) [155, 156] frontend and a traditional Gaussian mixture model (GMM) back-end [157, 158]. Version 2.0 appends deltas, double-deltas and log-energy to the CQCC features (in lieu of C0), and applies cepstral mean and variance normalisation. Classifier scores are computed as the log-likelihood ratio for the test utterance given bona fide and replayed speech models. This Chapter reports results only for the extended protocol baseline for which training and development are performed using pooled training and development dataset, referred

[4] http://www.asvspoof.org/data2017/baseline_CM.zip

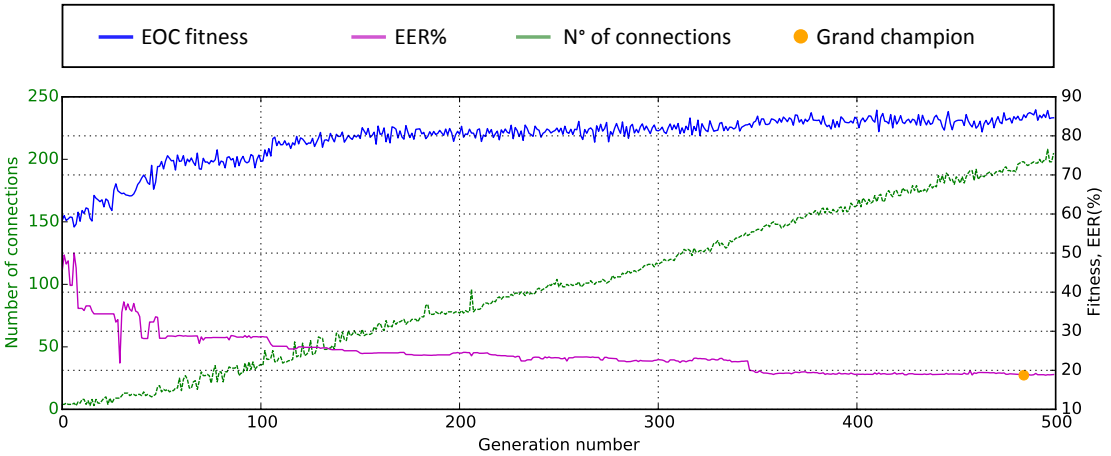


Fig. VII.3 *Evolution of 500 generations with an EOC fitness function with mini-batch training (EOC_m configuration). The upper blue profile shows the EOC-derived fitness of the fittest network in each generation. The highest fitness is obtained in generation 490. The green profile is the complexity (number of connections) in each network.The lower magenta profile is the EER of generation champions estimated using pooled training and development data. It reaches a minimum value in generation 484 (marked by an orange dot). This is the grand champion network that is chosen for testing on the evaluation set.*

to as train+dev. Baseline results for the extended protocol are presented to the top of Table VII.2.

VII.3.3 End-to-end anti-spoofing

All networks are configured according to the common setup described in Section VI.2 and as depicted in Fig. VI.5. Experiments were conducted with four different configurations comprising AUROC (Section VI.3.1) and EOC (Section VII.2.1) fitness functions with and without mini-batch training (Section VI.3.2). Configurations in which mini-batch is adopted are labelled m (see Table VII.2). Each configuration was run for 500 generations.

When applied, mini-batch training is performed with bona fide speech partitioned into four mini-batches of approximately 17 minutes each. Spoofed data is partitioned into three mini-batches, approximately 21 minutes each (see Section VII.2.2). The discrepancy between bona fide and spoofed speech is due to the greater variation in spoofed speech, the reliable modelling of which requires greater quantities of data in each batch.

Once the training of a generation is complete, the performance of networks for that generation is assessed according to the procedure described in Section VI.3.4, where *the whole training set* for this work corresponds to the pooled training and development data partitions (see Section VII.3.1).

VII.4 Experimental results

This section describes experimental results, starting with an illustration of the evolutionary behaviour of the end-to-end approach to spoofing detection and then an assessment of performance in terms of the EER. Also discussed here is the behaviour of the gate.

VII.4.1 Evolutionary behaviour

An illustration of the evolutionary behaviour of the end-to-end approach to spoofing detection is illustrated in Fig. VII.3. Two profiles show the evolution in EOC for the current mini-batch (top blue profile) and the number of network node connections (green dashed profile) of the highest-EOC network of the generation. The lower magenta profile shows the EER for the champion of each generation (the *generation champions*) estimated using train+dev data. The single network selected for the testing/evaluation is that which produces the lowest EER for the train+dev data (orange dot). This network is designated as the *grand champion* network.

The fitness is seen to increase gradually as the end-to-end approach to anti-spoofing learns to discriminate between bona fide and spoofed speech, gradually increasing network complexity as evolution proceeds. Improvements in fitness are largely accompanied by decreases in EER. After approximately 350 iterations, the EER seems to converge, with the best performing network being that from the 484th generation and having 198 connections.

VII.4.2 Spoofing detection performance

Results are presented in Table VII.2 for the baseline systems and the for the end-to-end system with AUROC and EOC fitness functions, with and without mini-batching (denoted by subscript m). Results for the EOC fitness function are either similar to or better than those for the AUROC fitness function. Mini-batching appears to offer inconsistent results for the AUROC fitness function; performance degrades for train+dev but improves for evaluation. For the EOC fitness function, improvements are consistent across the two data subsets.

Of particular interest is the stability or generalisation achieved by the end-to-end system. Performance for the baseline systems is seen to degrade substantially between the two sets (train+dev and evaluation). In contrast, the best results achieved with the end-to-end approach using the EOC fitness function and mini-batch training is not only substantially better, but also consistent across the two disjoint data sets (18%). Since results for the improved baseline Version 2.0 were not available at the time this work was carried out, the proposed end-to-end system (namely in its two EOC configurations) represented a substantial improvement over the original baseline system.

As already introduced in section VI.2, the gate acts to identify salient information in the network output, akin to an attention mechanism. It is stressed, though, that the gate operates on the *output* stream rather than on the *input* stream. Coupled with the recurrent nature of the network which maps inputs to outputs, this impedes a straightforward interpretation of its behaviour; it is difficult to interpret gate behaviour at the output with

Table VII.2: *End-to-end spoofing detection performance for the ASVspoof 2017 V2.0 database and extended protocol.*

	Train+Dev	**Eval**
Baseline V 1.0	0.1%	23.4%
Baseline V 2.0	2.5%	12.2%
AUROC	20.9%	28.2%
AUROC$_m$	27.4%	24.2%
EOC	20.3%	19.2%
EOC$_m$	18.7%	18.2%

respect to the acoustic stream at the input by just looking at the signal. Nevertheless, it is of interest to investigate its behaviour by observing some of the network outputs for bona fide and spoofed files, in the same fashion as done for ASV and as reported in Section VI.4.3.

Although observations made across only a handful of files are far from a statically-sound approach, Fig. VII.4 shows alternating periods of activity only in the first part of the output (although the scale of the picture makes them appear contiguous). In this case, with the output signal being still close to the input, it is easy to see how the first region corresponds to a very amplified or distorted transformation of the pre-speech part, while speech itself begins at about 0.25 seconds. This would seem to indicate that the network is learning something from non-speech intervals, *e.g.* perhaps information relating to the acoustic path between the loudspeaker and microphone. What is not clear, however, is exactly what the network is exploiting to produce higher scores for bona-fide trials and lower scores for spoofed speech trials. Further analysis would involve a deeper examination of how to link gate behaviour at the output to information at the input.

VII.5 Conclusions

This chapter reports the application of the end-to-end approach to the problem of spoofing detection. End-to-end techniques that avoid a reliance upon hand-crafted features are assumed to offer better potential for spoofing detection, and especially generalisation when the cues indicative of spoofing can vary considerably and are largely unpredictable in practice. Critical to performance is the proposed progress-rewarding fitness function which steers the evolutionary process progressively towards the reliable classification of a diminishing number of difficult trials. Coupled with a mini-batch training procedure, this particular quality of the proposed solution preserves generalisation.

Results for the ASVspoof 2017 Version 2.0 database show improvements to both generalisation and raw performance. Equal error rates for the end-to-end approach represent a 22% relative reduction compared to the baseline system Version 1.0, although current Version 2.0 yields the overall best result. Performance was not, however, the main focus of this work, which shows how the NeuroEvolution of augmenting topologies algorithm can be applied successfully to anti-spoofing operating directly on the raw audio wave-

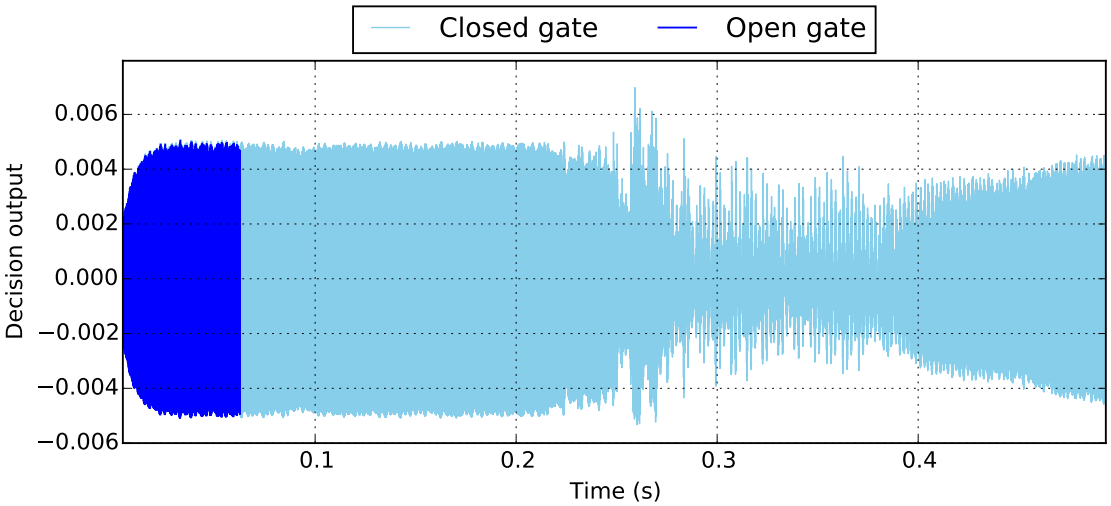


Fig. VII.4 *An example of network output from the grand champion of configuration EOCm. Dark and light coloured sections indicate gate-open and closed samples, respectively. It is stressed that the gate-open section is not contiguous, although it appears to be so given the high sample rate. Note how the output signal still bears resemblance to the input, with the first region being a very amplified or distorted version of the pre-speech part. Speech begins at around 0.25 seconds. Since the gate is closed for the remainder of the output, the score is only influenced by output samples non pertaining to speech. This opening-closing behaviour is consistent for all the trials processed by this network.*

form. The low footprint of the resulting grand champion network and the population in general, coupled with the time-domain nature of inputs and outputs means that it would be feasible to meaningfully interpret the output with respect to the acoustic stream at the input, provided it is done on a bigger scale involving a statistically significant number of networks and trials. This opens the doors for future findings that can actually explain *what makes it work* by directly studying the outputs of the networks. The findings of such a study, while left for further work, will help to determine precisely what information helps most to differentiate between bona fide and replayed speech.

Chapter VIII
Conclusions

This chapter reviews three years of research, summarising all the progress and contributions of the author during his PhD at NXP Semiconductors and setting the path for future work. ASV technologies are today gaining traction in industry and finding its way into everyday life. The contributions described in this thesis hopefully bring to attention how user convenience and low complexity (Chapter III) as well as security (Chapter IV) are important in real case scenarios. Work with highly experimental approaches (Chapters VI and VII) also shows that the research activity in a company is not limited to off-the-shelf component optimisation but also extends to blue-sky research.

This chapter summarises the achievements derived from the experiments carried out in the aforementioned chapters, and then concludes the thesis with global conclusions and views on future work.

VIII.1 From the laboratory into the wild

Work described in Chapter III is the least research-oriented and the most industrially flavoured contribution of this thesis. This work aimed to overcome an issue with speaker verification in a short-utterance recognition context. The need for several minutes of user speech for enrolment is impractical in most scenarios. The demand for such volumes of enrolment data is in stark contrast for the requirements of convenient, non-intrusive interaction and a plug-and-play setup, which are paramount in automatic speaker verification for smart device/home applications and in the Internet of Things (IoT) domain.

Results reproduced in Table VIII.1 show that it is possible to reduce the requirements in enrolment data of a text-dependent system thereby increasing usability, while introducting only modest degradation to performance. The proposed system is indeed a small modification of the HiLAM system described in Chapter II, consisting in a simplified 2-layer system. The baseline HiLAM system and the reduced 2-layer version are illustrated in Fig. VIII.1. Speaker enrolment is reduced from approximately five minutes to only three repetitions of a given sentence or pass-phrase, with the total training procedure lasting just a few seconds (accounting for 97% reduction in enrolment data).

Regardless of the admitted modest modification, experiments with the official RSR2015 database protocols show little to no performance drop, in some instances even yielding better results than the implemented HiLAM baseline as well as results in the litera-

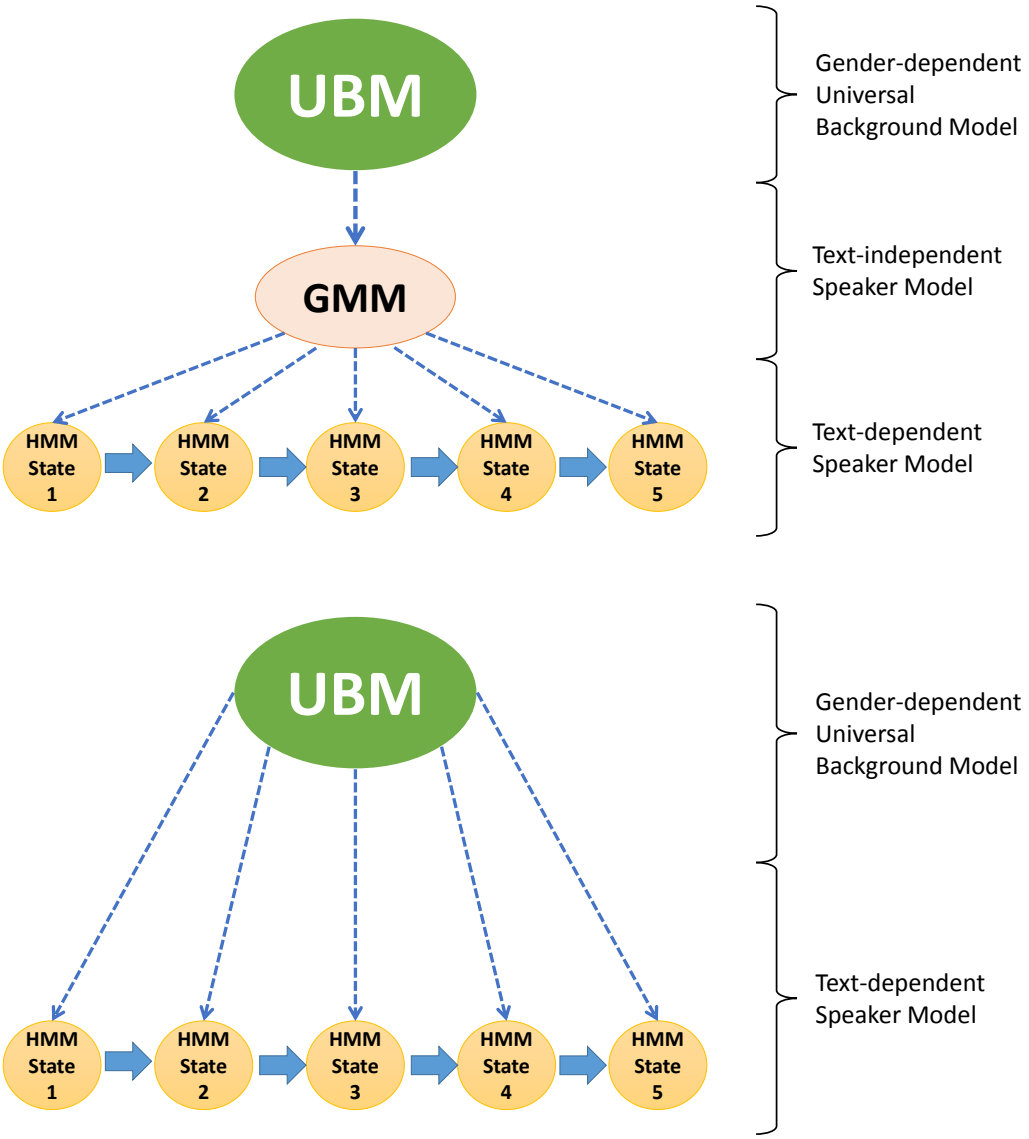


Fig. VIII.1 *Top: The original HiLAM system architecture reproduced from [37], first shown as Fig. II.4. Bottom: The simplified 2-layer architecture: text-dependent speaker models are adapted directly from the UBM, reproduced from Fig. III.2*

Table VIII.1: *Comparison of results for our implementation of the HiLAM system (3L) with original results reported in [59] and those obtained with the simplified system (2L). Reproduced from Table III.2)*

System	IC-Dev	TW-Dev	IW-Dev	IC-Eval	TW-Eval	IW-Eval
Larcher 3L [59]	1.43%	1.00%	0.20%	1.33%	0.66%	0.09%
Valenti 3L	1.63%	8.34%	0.78%	1.81%	7.54%	0.83%
Valenti 2L	1.84%	1.09%	0.32%	1.24%	0.52%	0.05%

ture [59]. This work demonstrated that reliable short-utterance text-dependent speaker verification, with little effort required from the user, is actually feasible.

VIII.2 Not all sentences are created equal

As a short sentence is obviously phonetically unbalanced, its choice is understandably crucial to the resulting ASV performance. Following the reduced-data work described in Chapter III, Chapter IV studies the inner text-dependent factors that can cause significant shifts in error rates when dealing with extremely short pass-phrases or commands in the order of one second.

Experiments were carried out using the HiLAM system and RSR2015 data part II, consisting of short commands designed for ASV systems used in smart home scenarios. The goal of the study was not just to report EER shifts in the score distributions and relate them to the text content; but also to universally classify a spoken password *strong* or *weak*. In order to do it, is key to prove that the ranking done with respect to one set of speakers still holds when applied to a second, independent set. The issue of spoken password strength is speaker-independent.

Fig. VIII.2 (a) and (b) show how different groups of 10 commands out of the 30 available can cause relative drops in the EER of up to 50% from the strongest to the weakest group; (c) and (d) show how well the ranking made with respect to one set of speakers (pictured by solid symbols) translates to an independent set (hollow symbols): the pertinence to the ranking is testified by the almost-monotonic trend of the hollow-symbol curves. The impact of the text content on a text-dependent system performance can be thus be assumed from one set of speakers and applied to another within the bounds of statistical significance, proving the consistency of *spoken password strength*.

This statistical analysis can be exploited in two ways: from a company point of view, it can help to choose the branded *one-size-fits-all* wake-up pass-phrase for services or devices, selecting the single pass-phrase with the highest level of discrimination among different speakers, *i.e.* "my voice is my passport" or "ok Google". From the user point of view, knowing that a certain pass-phrase may expose his or her device to impostures, can assist the user in avoiding weak, less secure pass-phrases, akin to warning mechanisms for written passwords often found on websites.

While the actual warning algorithm was not developed, the concept was registered as a patent [6]. This analytic study demonstrates that the text content has such a large impact

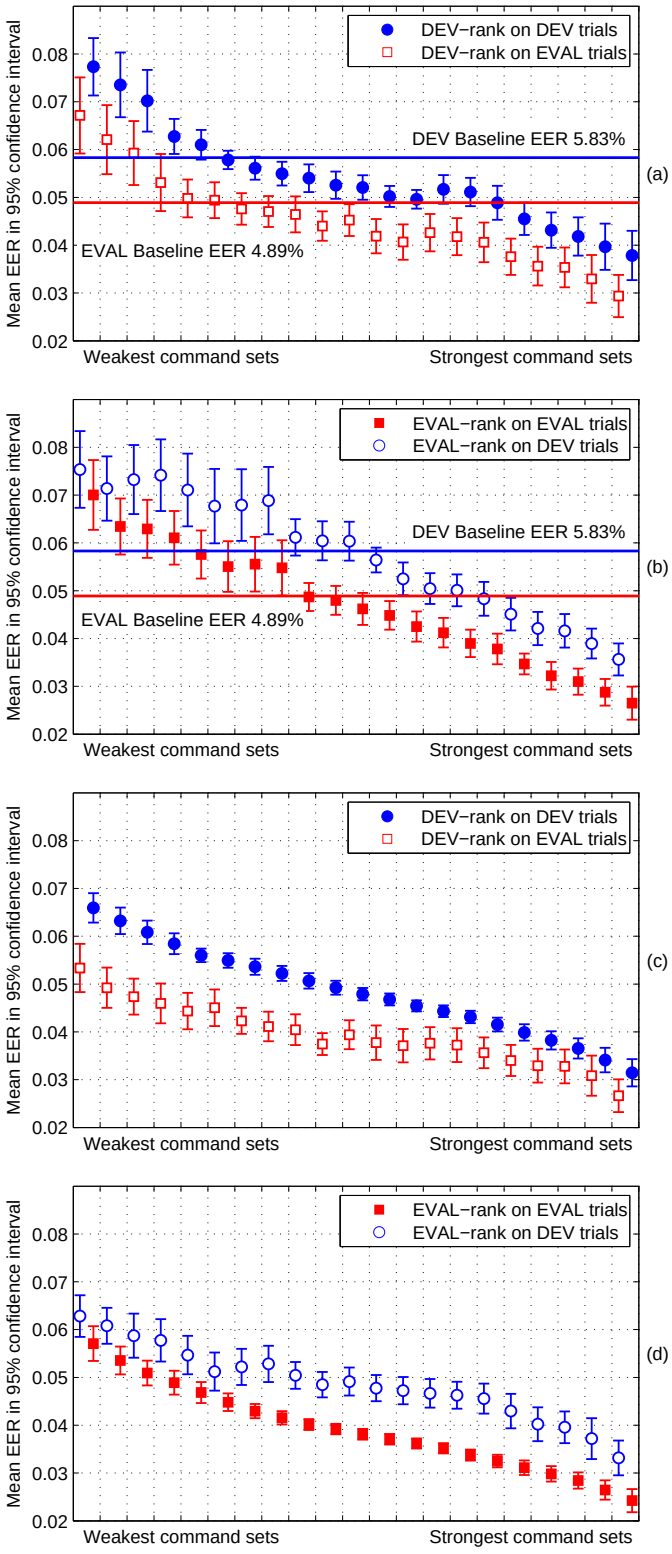


Fig. VIII.2 *ASV performance with (c,d) and without (a,b) text-dependent threshold adjustment. Reproduced from Fig. IV.2.*

on ASV system performance that it can be isolated and defined the speaker-independent metric of spoken password strength.

VIII.3 Truly end-to-end ASV

The contribution of Chapter VI represent one of the first attempts to design an approach to ASV that operates on the raw audio stream with dynamically optimised neural network topologies. The goal of this work was not to outperform existing approaches, but rather prove the feasibility of speaker modelling while leaving the question of topology optimisation to machine learning, *i.e.* without placing constraints on the features or on the neural architecture. Less complex topologies with a low memory footprint are well suited to embedded implementations, and also open the door to explain or interpret of *what makes them work*.

Both trends of replacing hand-crafted features with raw audio (or less processed inputs) and optimising the structure to the task at hand are present in the recent literature and are here applied for the first time in tandem for a truly end-to-end approach. The NeuroEvolution of Augmenting topologies (NEAT) algorithm is for the first time applied to ASV, which produces low-footprint text-independent speaker networks consisting of a few hundreds connections. These networks exhibit a layer-free topology, and every connection and weight was created or optimised through the evolutionary process depicted in Fig. VIII.3.

Given the highly experimental nature of the approach, experiments were performed on a relatively small scale. The downside of the approach is the very time-consuming training procedure which involves several generations of a population of networks that must be evolved separately for each speaker. The first set of experiments was performed using a proprietary NXP database collected from 10 speakers. Testing was conducted using the data of 2 speakers while the remaining data was used either as background data during training or as impostor data at testing. Fig. VIII.4 shows the evolution of the 2 speaker networks across 500 generations: decreases in EER values (right axis) for the training data correspond to decreases for test data. Results for both speakers are presented in Table VIII.2, along with comparisons to results for GMM- and NN-based baseline systems.

A second set of experiments was performed on a subset of the standard NIST SRE16 corpus. The SRE16 data contains only non-English language speech which proved to be a tough challenge for the end-to-end system, considering that all the background data available for training is English only. Results for the reduced protocol of 7 male speakers are reproduced in Table VIII.3 and are compared to those obtained for an i-vector baseline (labelled ICMC). In both sets of experiments the potential of the end-to-end approach is evident. Both sets of experiments depicted an approach which is still in its infancy. It is undeniable, though, especially by observing the descent of the EER curves in Fig. VIII.4 and model ID 1078 in Table VIII.3 that the system is learning discriminative information about the target speaker. It is therefore possible to perform speaker verification in the time domain, on the raw audio signal. While the approach is promising, there is obvious room for improvement in several aspects for it to be exploited to the full extent.

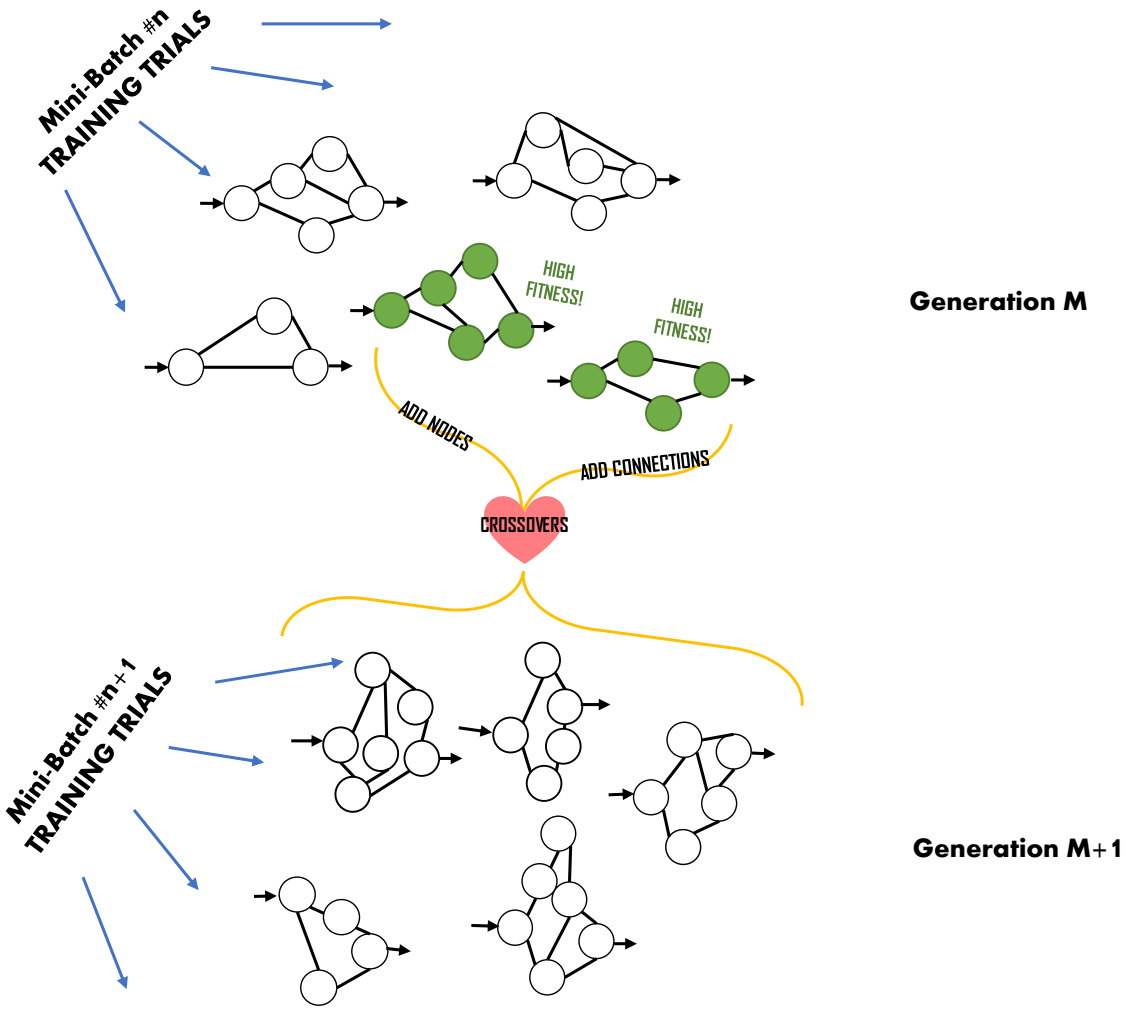


Fig. VIII.3 *An illustration of 2 subsequent training iterations, reproduced from Fig. VI.7*

VIII.4 Truly end-to-end anti-spoofing

The end-to-end approach described in Chapter VI is discriminative in nature. Each champion network, which required its own evolutionary process, was used to discriminate between the target speaker and several impostors, with the latter class being way more variate and data-plentiful than the former. Although not strictly the focus of this thesis, anti-spoofing presented itself as a potentially ideal fit for the NEAT-based approach. First, its binary discriminative nature could be put to use to separate genuine and spoofed speech encompassing multiple speakers. Second, anti-spoofing is a relatively young domain and the search for the appropriate features is still ongoing, implying that operation upon raw-audio may offer rapid returns. Third, the learning would require just one evolutionary process, making experimentation on official protocols computationally feasible.

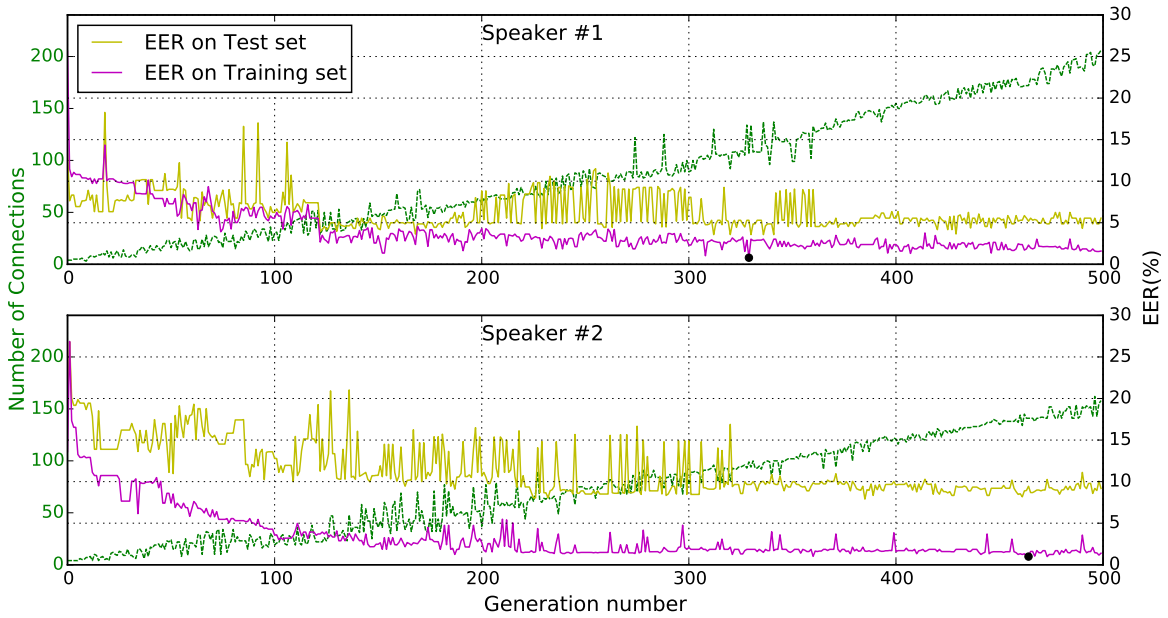


Fig. VIII.4 *Number of connections (green dashed profiles) and equal error rate (EER) of the first 500 generation champions for target speakers 1 (top) and 2 (bottom). EER profiles are shown for training data (magenta/dark) and testing data (yellow/light profiles). Black dots signify the grand champion, chosen according to the lowest-EER on the full training set. Reproduced from Fig. VI.9.*

Table VIII.2: *Results for the baseline systems and end-to-end system in terms of EER for the training and test set for the two target speakers. Reproduced from Table VI.1*

	Speaker #1		**Speaker #2**	
	Training	Test	Training	Test
GMM-UBM	0%	9.5%	0%	6.9%
CNN$_{MFCC}$	0.8%	6.8%	0%	2.5%
CNN$_{logmel}$	0.4%	5.8%	0%	1.0%
RC-DNN$_{MFCC}$	0.6%	9.0%	0.2%	1.7%
RC-DNN$_{logmel}$	0.8%	6.5%	0.2%	1.2%
End-to-end	0.8%	5.3%	1.0%	9.4%

Table VIII.3: *Results on NIST SRE16 development set data for the end-to-end system and the ICMC system reported in [10]. Results between parentheses are for champion networks chosen a posteriori and are just reported for observations. Reproduced from Table VI.4.*

Model ID	Language	E2E	ICMC
1008	Mandarin	46.5 (29.6)%	6.7%
1011	Mandarin	29.4 (15.2)%	0.3%
1036	Mandarin	23.0 (15.8)%	9.4%
1050	Mandarin	24.6 (18.0)%	8.1%
1039	Cebuano	16.4 (14.1)%	12.9%
1043	Cebuano	25.3 (**15.4**)%	16.9%
1078	Cebuano	**21.6** (**20.0**)%	36.2%
Average 1	Mandarin	30.9 (19.7)%	6.1%
Average 2	Cebuano	**21.1** (**16.5**)%	22.0%

Table VIII.4: *End-to-end spoofing detection performance for the ASVspoof 2017 V2.0 database and extended protocol. Reproduced from Table VII.2.*

	Train+Dev	Eval
Baseline V 1.0	0.1%	23.4%
Baseline V 2.0	2.5%	12.2%
AUROC	20.9%	28.2%
AUROC$_m$	27.4%	24.2%
EOC	20.3%	19.2%
EOC$_m$	18.7%	18.2%

Chapter VII reported the application of the NEAT-based end-to-end approach to replay attack detection and its assessment using official ASVspoof 2017 database protocols. For the task, the system was equipped with a new progress-rewarding fitness function which steers the evolutionary process progressively towards the reliable classification of a diminishing number of difficult trials. Although this fitness function, named *ease of classification* (EOC), was initially developed for ASV, it has proven successful in the case of anti-spoofing.

Results for the ASVspoof 2017 Version 2.0 database are reproduced in Table VIII.4 which shows EERs for the official ASVSpoof 2017 baseline versions 1.0 and 2.0 compared to 4 different configurations of the E2E approach. The configurations are named after the fitness function used, the subscript m indicating the use of mini-batch during training. Although no configuration achieves better result than the current baseline system, equal error rates for the end-to-end approach represent a 22% relative reduction compared to version 1.0, which was the only baseline when the work in Chapter VII was carried out.

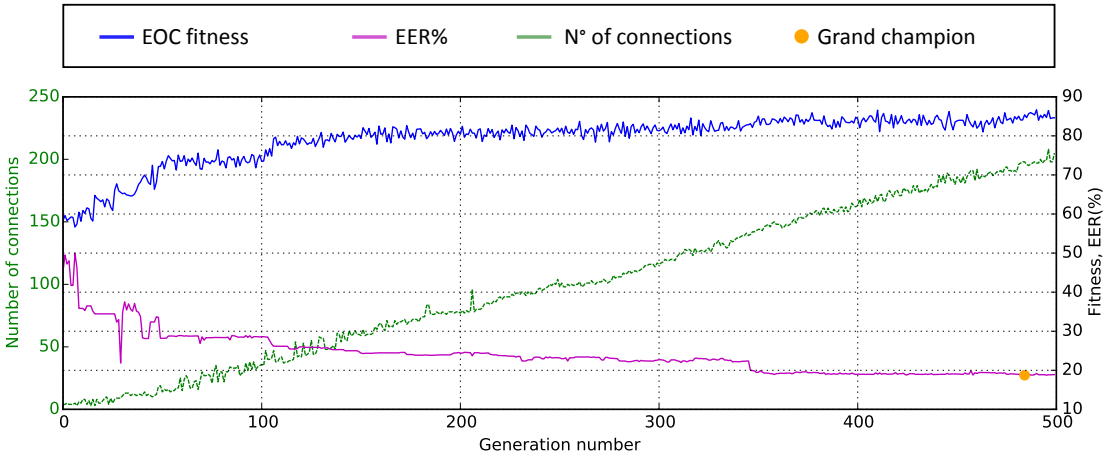


Fig. VIII.5 *Evolution of 500 generations with an EOC fitness function with mini-batch training (EOC_m configuration), reproduced from Fig. VII.3*

The graph in Fig. VIII.5 clearly shows how the end-to-end system learns through mini-batches of raw audio inputs: as the highest EOC value increases across generations, the EER on the whole "train+dev" data decreases from 50% to the 18.7% reported in Table VIII.4. Moreover, the performance obtained for the evaluation set is consistent with the EER on the evaluation set. This confirms that it is possible to perform spoofing detection on the raw audio waveform with dynamically optimised neural architectures.

VIII.5 Closing thoughts and future work

This 3 years of research allowed the author to contribute to efficiency and security for ASV in embedded systems in different ways and through different means. Efficiency for the user, which is freed from minute-long initialisation of his or her ASV-enabled device; efficiency for the company, which is one step closer to embedding topology-optimised small-footprint neural networks (that require no preprocessing of the input speech) in low-resource devices.

In a world where "your voice is your password", security comes from knowledge of which spoken passwords are the most secure, whether customised by the user or fixed by the developer for everybody. In the speaker verification domain, security goes hand in hand with spoofing countermeasures; while not the focus of this thesis, anti-spoofing has proven to be a fitting terrain to experiment with NeuroEvolution of augmenting topologies. The blue-sky research done within this alternative approach to deep, complex neural network architectures occupied the latter half of the PhD, and resulted in one of the first applications of augmenting topologies on raw audio for both ASV and anti-spoofing.

While filled with interesting findings, the author is aware that the work carried out during his time as a PhD student left several questions unanswered. Concerning the HiLAM simplification work of Chapter III, there is still an unresolved problem with Target-Wrong trials for the 3-layer system (see Table VIII.1). It doesn't affect the 2-layer variant which is the focus of the work, but the issue should be nevertheless investigated

since it is not present in the literature [59] and the implemented 3-layer system is the basis for the 2-layer version. It would also be interesting to know if reducing the speaker-dependent data actually leaves the proposed 2-layer system very weak to noise or spoofing compared to the 3-layer variant.

Regarding the work on spoken password strength, an *a priori* warning system for supposedly weak passwords could be implemented, as introduced in Section VIII.2. This warning feature would require prior knowledge on the strength of any possible spoken pass-phrase, which could be obtained by studying the strength of smaller linguistic blocks, such as phonemes. By experimenting on large-scale phoneme-level labeled data, an effective measurement of password strength could be given at choice time.

The blue-sky research work that led to the end-to-end system used for the work in Chapters VI and VII and here resumed in Sections VIII.3 and VIII.4 has understandably the largest room for improvement and future work.

The grand champion selection policy (see Section VI.3.4) needs to be improved. Though the author was already aware of the issue when experimenting on the NXP database, experiments with the NIST SRE16 corpus showed how suboptimal the grand champion selection is, especially when the training data presents some *global shift* in its characteristics (language mismatch, in this case). The selection policy could be improved at first with simple requirements for the selected grand champion such as minimum generation number and minimum complexity, and also using a fusion of evaluation metrics (AUROC, EER, EOC or others).

Observing the sample-by-sample network activity is of key importance to achieve true explicability which in this thesis was just theoretically rendered feasible by the relatively low number of connections of the grand champion networks. It would be interesting to collect statistics on the output scores and finding an interpretable visual representation, akin to t-Distributed Stochastic Neighbor Embedding (t-SNE) [159], which maps certain phenomena between input, network inner behaviour and output. Monitoring the gate activity on a large number of trials and searching for particular trends would also contribute to explicability, since both the output unit and the gate learn their behaviour with respect to each other and, in the end, only the gate-open samples contribute to the final score.

Appendix A
Published work

Audio Engineering Society

Convention Paper 9844

Presented at the 143rd Convention
2017 October 18–21, New York, NY, USA

A Simplified 2-Layer Text-dependent Speaker Authentication System

Giacomo Valenti[1,2], Adrien Daniel[1], and Nicholas Evans[2]

[1]*NXP Software, Mougins, France*
[2]*EURECOM, Biot, France*

Correspondence should be addressed to Giacomo Valenti (giacomo.valenti@nxp.com)

ABSTRACT

This paper describes a variation of the well-known HiLAM approach to speaker authentication which enables reliable text-dependent speaker recognition with short-duration enrollment. The modifications introduced in this system eliminate the need for an intermediate text-independent speaker model. While the simplified system is admittedly a modest modification to the original work, it delivers comparable levels of automatic speaker verification performance while requiring 97% less speaker enrollment data. Such a significant reduction in enrollment data improves usability and supports speaker authentication for smart device and Internet of Things applications.

1 Introduction

The rapidly-growing smart device market and the explosion of the Internet of Things (IoT) has fueled the need for low footprint and efficient speaker authentication solutions, e.g. [1]. Unfortunately, many approaches to Automatic Speaker Verification (ASV) place unrealistic demands on enrollment and recognition/test data [2]. The need for anything more than a few seconds of speech impacts on usability and creates resistance among mass-market users.

ASV research has largely been driven by the Speaker Recognition Evaluations (SREs) administered by the US National Institute of Standards and Technology (NIST)[1]. These evaluations have typically focused on enrollment and testing with a duration in the order of a few minutes. While the SREs have stimulated tremendous progress over the last two decades, today's state-of-the-art speaker verification technology is often ill-suited to authentication applications which demand reliable recognition using utterances with a duration in the order of a few seconds [3, 4, 5]. With a clearly different use case scenario, the NIST SREs have also focused on text-independent recognition, whereas short-duration recognition generally calls for text-dependent operation.

State-of-the-art i-Vector and probabilistic linear discriminant analysis (PLDA) techniques are difficult to apply in text-dependent tasks [6, 7, 8] unless training data is plentiful [9] and unless impostor trials involve matching text [10]. Studies reported in [11, 12, 13, 14] demonstrated that joint factor analysis (JFA) systems can work well with little enrollment data, however, even under those conditions, both JFA and PLDA still rely on prior knowledge of the text content.

Initiatives dedicated to furthering progress in text-

[1] https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016

dependent recognition have gathered pace in recent years, prominent examples being the release of the RSR2015 [15] and RedDots [16] databases and associated evaluation campaigns. The RSR2015 database was furthermore introduced together with a baseline ASV system referred to as HiLAM (Hierarchical multi-Layer Acoustic Model) [6]. It involves a 3-layer approach to text-dependent speaker modeling. The HiLAM system is today a reference approach. Even it, though, is ill-suited to our target application since it learns an intermediate text-independent speaker model which in turn requires significant speaker enrollment data.

We have thus sought to develop an alternative to the HiLAM system which reduces demands on enrollment data for a short-duration, text-dependent speaker authentication application. Since the target application is text-dependent, the aim is to dispense with text-independent enrollment entirely. While an admittedly modest modification to the original work, the result is a simpler two-layer approach which achieves comparable ASV performance with a dramatic reduction in the need for enrollment data.

The remainder of this paper is organized as follows. Section 2 describes the RSR2015 database which was used for all experimental work reported herein. The original HiLAM baseline system is summarized in Section 3 whereas modifications to support short-duration speaker enrollment are presented in Section 4. A thorough comparison of the two systems performed using the standard RSR2015 evaluation protocol is presented in Section 5. Conclusions are presented in Section 6.

2 Database and Protocols

Almost all experimental work undertaken using the HiLAM system [6, 17, 18] is performed using the RSR2015 database [15]; the two were released almost in tandem and the database is distributed with protocols suited to the assessment of HiLAM-based text-dependent speaker verification systems. The RSR2015 database is one of the most versatile and comprehensive databases for such research. One particular aspect of RSR2015 which makes it better suited to this work than the more recent RedDots [19] successor is the particular speaker/part/session structure illustrated in Fig. 1. This is described below.

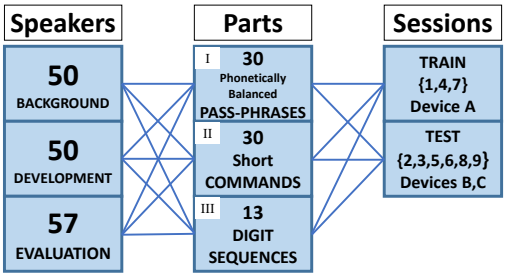


Fig. 1: *RSR2015 database partition for male speakers. The partition is identical for female speakers but with only 43 speakers in the evaluation set.*

2.1 Database

RSR2015 contains speech data collected from both male and female speakers and is partitioned into 3 evenly-sized subsets whose usual purpose is for background modeling, experimental development and evaluation. Each subset is comprised of 3 parts: phonetically-balanced sentences (part I), short commands (part II) and random digits (part III). Each part contains data collected in one of nine sessions. Three of these sessions are reserved for training while the remaining six are set aside for testing. The three training sessions are recorded using the same smart device (i.e. the same mobile phone or tablet) whereas the six testing sessions are recorded using two different smart devices.

Since our target application relates to short-duration pass-phrases, all experimental work reported in this paper was performed using part I data consisting of phonetically-balanced sentences. These are the same 30 Harvard sentences used in the collection of the better-known TIMIT database [6] which were designed to give a broad coverage of phonemes in the English language.

2.2 Training Protocol

Data reserved for background modeling is disjoint from training and testing data; there is no overlap in terms of speakers or sentences. Second-layer HiLAM models (GMMs) are trained with data from all three training sessions and all 30 sentences, totaling 90 utterances. Third-layer HiLAM models (HMMs) are trained with the three training utterances corresponding to each specific sentence (30 models each adapted from the second-layer model with three repetitions of each sentence).

Table 1: *The four different trial types used to assess the performance of a text-dependent speaker verification system. They involve different combinations of matching speakers and text.*

Trial Type	Speaker Match	Text Match
Target-Correct (TC)	Yes	Yes
Target-Wrong (TW)	Yes	No
Impostor-Correct (IC)	No	Yes
Impostor-Wrong (IW)	No	No

Initial experiments reported in this paper were performed using the standard protocols which are distributed with the RSR2015 database. However, since the goal of the work reported here is to reduce the quantity of data (number of utterances) needed for speaker enrollment, subsequent experiments were performed with subsampled versions of the standard protocols. As described later, the amount of data used for the learning of second-layer models is then either reduced (3-layer system with protocol sub-sampling) or eliminated entirely.

2.3 Testing Protocols

Test results reflect recognition performance estimated from a large number of single-utterance trials. Testing protocols used for all experiments are the standard part I testing protocols distributed with the RSR2015 database. All relate to one of the four trial types illustrated in Table 1. Any given trial involves either a target (model and test utterance correspond to the same speaker) or an impostor (model and test utterance correspond to different speakers). In addition, the text content either matches across model and test utterance (correct) or is different (wrong). This leads to three testing conditions which assess performance combining target-correct trials with trials of **one** mismatching combination: target-wrong, impostor-correct or impostor-wrong (note that target-wrong is therefore considered an impostor trial). The number of trials for each type in the standard RSR2015 protocols is illustrated in Table 2 for development and evaluation sets. The number of trials for each testing condition is TC+TW, TC+IC and TC+IW respectively. Finally, performance is expressed in terms of the equal error rate (EER).

Table 2: *Number of trials for Part I of the RSR2015 database for each of the four trial types illustrated in Table 1 and for development (Dev) and evaluation (Eval) subsets.*

Speaker-Text	Dev	Eval
Target-Correct (TC)	8,931	10,244
Target-Wrong (TW)	259,001	297,076
Impostor-Correct (IC)	437,631	573,664
Impostor-Wrong (IW)	6,342,019	8,318,132

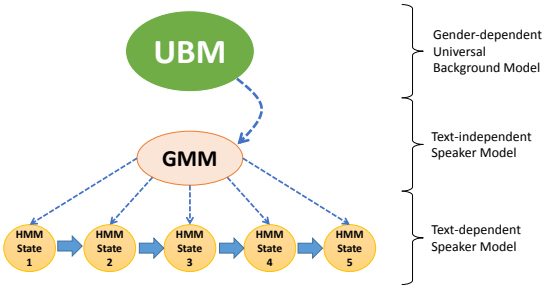


Fig. 2: *The original HiLAM system architecture reproduced from [17].*

3 The HiLAM Baseline

This section describes the original HiLAM architecture and essential elements of the basic algorithm. Maximum a posteriori (MAP) adaptation [20] is given particular attention; its optimization is fundamental to the simplified version of HiLAM presented later. Also presented are results for our specific implementation assessed using the RSR2015 database.

3.1 Architecture and Algorithm

The HiLAM system is a flexible, efficient and competitive approach to text-dependent automatic speaker verification. The architecture is illustrated in Fig. 2 and is composed of three distinct layers. They represent (i) a gender-dependent universal background model (UBM), (ii) a text-independent speaker model and (iii) a text-dependent speaker model. The first and second layers take the form of Gaussian mixture models (GMMs) whereas the third layer is a hidden Markov model (HMM).

The UBM is trained according to a conventional maximum likelihood / expectation maximization criterion [21]. The second layer text-independent speaker

model is derived from the UBM via MAP adaptation; this procedure is described in detail below. Different third-layer text-dependent speaker models are then learned for each sentence or pass-phrase. These take the form of 5-state, left-to-right HMMs. Each state of the HMM is initialized with the second layer text-independent GMM of the corresponding speaker and then adapted with several iterations of Viterbi realignment and retraining [22]. Each HMM therefore captures both speaker characteristics in addition to the time-sequence information which characterizes the sentence or pass-phrase. Full details of the HiLAM system in addition to the training and testing procedures can be found in [6].

3.2 MAP Adaptation

MAP adaptation is used to obtain the second-layer GMM from the first-layer UBM. A fundamental parameter of the MAP algorithm which governs the degree of adaptation is the so-called relevance factor, τ. Together with a probabilistic count of new data n_i for each Gaussian component i, it is used to determine an adaptation coefficient given by:

$$\alpha_i^\rho = \frac{n_i}{n_i + \tau^\rho} \tag{1}$$

where $\rho \in \{\omega, \mu, \sigma\}$ indicates the relevance factor for the weight, mean or variance parameters of the GMM. The adaptation coefficients are then used to obtain the new weight, mean and variance estimates according to:

$$\hat{\omega}_i = [\alpha_i^\omega n_i/T + (1-\alpha_i^\omega)\omega_i]\gamma \tag{2}$$

$$\hat{\mu}_i = \alpha_i^\mu E_i(x) + (1-\alpha_i^\mu)\mu_i \tag{3}$$

$$\hat{\sigma_i^2} = \alpha_i^\sigma E_i(x^2) + (1-\alpha_i^\sigma)(\sigma_i^2 + \mu_i^2) - \mu_i^2 \tag{4}$$

where each equation gives a new estimate from a combination of the respective training data posterior statistics with weight α and prior data with weight $(1-\alpha)$. T is a normalization factor for duration effects; γ is a scale factor which ensures the unity sum of weights. $E_i(x)$ and $E_i(x^2)$ are the first and second moments of posterior data whereas μ_i and σ_i^2 are the mean and variance of prior data, respectively [23].

In our experiments, each stage of adaptation is performed with a common value of τ, and hence α, for Equations 2, 3 and 4; the use of different values does not lead to better performance. Two distinct relevance factors are used at each stage, however: (i) for the adaptation of the UBM to the GMM, τ_1 and (ii) for the adaptation of the GMM to the HMM, τ_2. The first relevance factor, τ_1, acts to balance the contribution of the UBM and speaker-specific adaptation data to the parameters of the new speaker model, while the second, τ_2, controls adaptation between the text-independent and text-dependent speaker models.

3.3 Configuration and Performance

Silence removal is first applied to raw speech signals sampled at 16 kHz. This is performed according to ITU-T recommendation P.56[2] which specifies an active speech level of 15.9 dB. In practice this results in the removal of approximately 36% of the original data. The remaining 64% is then framed in blocks of 20ms with 10ms overlap. The feature extraction process is standard and results in 19 static Mel frequency cepstral coefficients (MFCC) without energy (C0). These are appended with delta and double-delta coefficients resulting in feature vectors of 57 dimensions.

The number of Gaussian components is empirically optimized. The literature shows that higher values (512-2048) are often used for text-independent tasks [24, 23] or with systems based on i-Vector and PLDA techniques [10, 25]. In contrast, lower values (128-256) are typically used in text-dependent tasks and techniques such as HiLAM [26, 17]. We obtained the best performance with 64 Gaussian components.

Results for our implementation of the HiLAM baseline are presented in Table 3 alongside those presented in the original work [27]. Results are presented for male speakers only and for the most challenging IC impostor condition. While results for our system are worse than those in the original work, performance is still respectable, with EERs of less than 2% for both development and evaluation subsets.

4 Simplified HiLAM

Described in this section are experiments which assess the necessity of text-independent enrollment and a number of modifications to the original HiLAM baseline system which enable competitive performance with

[2] http://www.itu.int/rec/T-REC-P.56-201112-I/en

Table 3: *Comparison of results for our implementation of the HiLAM system with original results reported in [27]. Results shown for male speakers in part I of the RSR2015 database and for the IC impostor condition.*

Subset	Our Implementation	Larcher et al. [27]
Development	1.63%	1.43%
Evaluation	1.81%	1.33%

greatly reduced durations of speaker enrollment data. Among these modifications is the reduction of the 3-layer approach to only two layers and associated re-optimization. The new system learns text-dependent speaker models using only three training utterances.

4.1 Enrollment Demands

The HiLAM system is well-suited to applications involving both text-independent *and* text-dependent speaker recognition scenarios. Satisfactory performance in these two scenarios calls for a large amount of training data; the original HiLAM system reported in [6] used 90 utterances for training middle layer text-independent speaker models.

The need for such an amount of enrollment data can be impractical, if not unusable in many cases such as smart device and Internet of Things applications. This paper extends the past work to address an exclusive text-dependent scenario which demands far less enrollment data. The following describes a simplified approach which eliminates the middle layer entirely and which delivers competitive text-dependent recognition with only three training utterances with only modest performance degradation.

4.2 Necessity of Text-Independent Enrollment

Our optimization of the original HiLAM system showed that the best performance is delivered with comparatively higher and lower values of τ_1 and τ_2 respectively (see Equation 1). This finding indicates that only modest adaptation is applied between layers 1 and 2, whereas more significant adaptation is applied between layers 2 and 3. This then calls into question the real need for text-independent enrollment or, in other words, the real need for the middle layer.

Table 4: *Performance for different durations of 2nd layer text-independent training. The last row shows results for the simplified HiLAM system with no text-independent training. Results shown for the RSR2015 development set and for the IC condition.*

	Number of utterances	EER
3-Layer	90+3	1.63%
	60+3	1.66%
	30+3	1.62%
	3	2.33%
2-Layer	3	1.84%

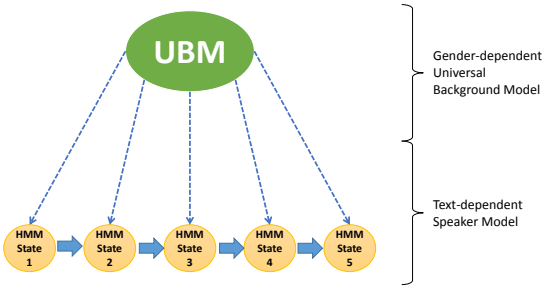


Fig. 3: *The simplified 2-layer architecture: text-dependent speaker models are adapted directly from the UBM.*

In order to assess the necessity of text-independent enrollment, we conducted a sequence of experiments in which the number of text-independent utterances used for layer-two training was successively subsampled from 90 to 60 and then 30 by taking 2 and 1 training sessions out of 3, respectively. Results are illustrated in Table 4. They show that performance remains unchanged as the quantity of text-independent enrollment data is reduced from 90 to 30 utterances. This finding suggests that text-independent enrollment may be unnecessary when the recognition task is ultimately text-dependent.

4.3 Layer Reduction

Given the observations reported above, we decided to assess performance when the middle layer, text-independent enrollment is dispensed with entirely. Speaker enrollment is then performed in text-dependent fashion exclusively as illustrated in Fig. 3. Each state

Table 5: *Comparison of results for the original work [27] and those obtained with the simplified system reported in this paper. Results for male speakers in part I of the RSR2015 database. (Results for each condition correspond to their combination with TC trials.)*

System	IC-Dev	TW-Dev	IW-Dev	IC-Eval	TW-Eval	IW-Eval
Larcher 3-Layer	1.43%	1.00%	0.20%	1.33%	0.66%	0.09%
Valenti 2-Layer	1.84%	1.09%	0.32%	1.24%	0.52%	0.05%

of the HMM speaker model is now initialized using the UBM instead of the speaker-specific text-independent GMM. Adaptation is otherwise the same as before and performed using the same three utterances of the same sentence. The number of Gaussian components (64) is left unchanged from the 3-layer implementation (see Section 3.3) and the single remaining relevance factor τ (3) is set to the same value of τ_2 (see Section 3.2). These parameters were found to be optimal in the case of the simplified system.

Results are illustrated in the last row of Table 4. Performance degrades slightly, from an EER of 1.6% for the baseline 3-layer system to 2.3% when enrollment is performed with only 3 speaker-specific utterances. Performance for the reduced 2-layer system improves slightly to 1.8%. Despite a reduction in enrollment data in the order of 97%, the increase in error rate is only 0.2%. Such a compromise between performance and usability would be quite acceptable in many practical scenarios.

5 Evaluation Results

Results presented above relate to the development set and the IC condition only. Presented in this section is a full performance comparison of the original HiLAM approach in [27] to the simpler 2-layer system presented in this paper using the full RSR2015 development and evaluation sets, including the three different test conditions, namely IC, TW and IW.

Results are illustrated in Table 5. The first row indicates the specific test condition for development (dev) and evaluation (eval) sets. Results presented in the original work [27] are illustrated in the second row whereas those for the new 2-layer system are presented in the third row. They correspond respectively to the full enrollment condition (90 text-independent utterances for layer 2 and 3 text-dependent utterances for layer 3) and the reduced enrollment condition (3 text-dependent utterances only). These results confirm the findings presented above, namely that significant improvements to usability can be delivered by reducing the demand for enrollment data with only modest increases in error rates. Both systems achieve better performance for the evaluation set than for the development set. While this finding is counter-intuitive, it is consistent with other results in the literature, e.g. [15, 17, 26, 27, 28], one possible explanation for which is differences in the distributions of recording devices across the two subsets.

Compared to the original work, performance for the 2-layer system deteriorates for the development set. In contrast, performance for the evaluation set improves. This result is particularly encouraging. The drop from 1.33% to 1.24% corresponds to a 7% relative reduction in the EER and comes with the same 97% reduction in demand for enrollment data. This is a significant improvement to usability in the case of text-dependent recognition.

6 Conclusions

This paper proposes a simplified version of the HiLAM approach to text-dependent automatic speaker verification in order to reduce the demand for speaker enrollment data. Many practical use case scenarios such as speaker authentication for smart device/home applications and those in the Internet of Things (IoT) domain call for enrollment with only a small number of passphrase repetitions. Experimental work presented in the paper questions the necessity of text-independent enrollment used in the conventional HiLAM system in the case that the ultimate recognition task is text-dependent in nature. Results produced using a publicly available, standard database and protocols show that text-independent, middle-layer enrollment impacts unnecessarily on usability. The paper shows that the middle layer of the HiLAM system and, hence, text-independent enrollment can be dispensed with entirely. Speaker enrollment is then performed using only three

repetitions of a given sentence or pass-phrase in a simplified two-layer approach. Since the collection of enrollment data is one of the most invasive and inconvenient tasks from the end user perspective, the usability of the new system improves greatly on the previous 3-layer HiLAM baseline system. The proposed approach, admittedly a modest modification of the original system, delivers largely comparable levels of automatic speaker verification performance with a 97% reduction in enrollment data.

References

[1] Lee, K. A., Ma, B., and Li, H., "Speaker Verification Makes Its Debut in Smartphone," in *IEEE SLTC Newsletter*, February 2013.

[2] Martınez, P. L. S., Fauve, B., Larcher, A., and Mason, J. S., "Speaker Verification Performance with Constrained Durations," in *International Workshop on Biometrics and Forensics (IWBF)*, IEEE, 2014.

[3] Kenny, P., Dehak, N., Ouellet, P., Gupta, V., and Dumouchel, P., "Development of the primary CRIM system for the NIST 2008 speaker recognition evaluation." in *INTERSPEECH*, pp. 1401–1404, 2008.

[4] Fauve, B. G., Evans, N. W., and Mason, J. S., "Improving the performance of text-independent short duration SVM-and GMM-based speaker verification," in *Odyssey Speaker and Language Recognition Workshop*, pp. 18–25, 2008.

[5] Poddar, A., Sahidullah, M., and Saha, G., "Performance comparison of speaker recognition systems in presence of duration variability," in *2015 Annual IEEE India Conference (INDICON)*, pp. 1–6, IEEE, 2015.

[6] Larcher, A., Lee, K., Ma, B., and Li, H., "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, 60, pp. 56–77, 2014.

[7] Aronowitz, H., "Voice Biometrics for User Authentication," in *Afeka-AVIOS Speech Processing Conference 2012*, 2012.

[8] Sahidullah, M. and Kinnunen, T., "Local spectral variability features for speaker verification," *Digital Signal Processing*, 50, pp. 1–11, 2016.

[9] Stafylakis, T., Kenny, P., Ouellet, P., Perez, J., Kockmann, M., and Dumouchel, P., "I-Vector/PLDA variants for text-dependent speaker recognition," *CRIM Technical Report*, 2013.

[10] Stafylakis, T., Kenny, P., Ouellet, P., Perez, J., Kockmann, M., and Dumouchel, P., "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *INTERSPEECH*, pp. 3651–3655, 2013.

[11] Kenny, P., Stafylakis, T., Ouellet, P., and Alam, M. J., "JFA-based front ends for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1705–1709, IEEE, 2014.

[12] Kenny, P., Stafylakis, T., Alam, J., and Kockmann, M., "JFA modeling with left-to-right structure and a new backend for text-dependent speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4689–4693, IEEE, 2015.

[13] Kenny, P., Stafylakis, T., Alam, J., Ouellet, P., and Kockmann, M., "Joint Factor Analysis for Text-Dependent Speaker Verification," in *Odyssey Speaker and Language Recognition Workshop*, pp. 1705–1709, 2014.

[14] Stafylakis, T., Kenny, P., Alam, M. J., and Kockmann, M., "Speaker and channel factors in text-dependent speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24, pp. 65–78, 2016.

[15] Larcher, A., Lee, K., Ma, B., and Li, H., "RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases." in *INTERSPEECH*, pp. 1580–1583, 2012.

[16] Lee, K. A., Larcher, A., Wang, G., Kenny, P., Brummer, N., and others, "The RedDots data collection for speaker recognition," in *INTERSPEECH*, pp. 2996–3000, 2015.

[17] Larcher, A., Lee, K., Ma, B., and Li, H., "RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases," in *INTERSPEECH*, pp. 1580–1583, 2012.

[18] Larcher, A., Lee, K. A., Ma, B., and Li, H., "Imposture classification for text-dependent speaker

verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[19] Lee, K. A., Larcher, A., Wang, G., Kenny, P., Brümmer, N., et al., "The RedDots Data Collection for Speaker Recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[20] Lee, C.-H. and Gauvain, J.-L., "Speaker adaptation based on MAP estimation of HMM parameters," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pp. 558–561, IEEE, 1993.

[21] Bishop, C. M., *Pattern recognition and machine learning*, Information science and statistics, Springer, 2006.

[22] Rodríguez, L. J. and Torres, I., "Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition," in *Pattern Recognition and Image Analysis*, pp. 847–857, Springer, 2003.

[23] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B., "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, 10(1-3), pp. 19–41, 2000.

[24] Bimbot, F., Bonastre, J.-F., Fredouille, C., and others, "A tutorial on text-independent speaker verification," *EURASIP journal on applied signal processing*, 2004, pp. 430–451, 2004.

[25] Larcher, A., Bousquet, P.-M., Lee, K. A., Matrouf, D., Li, H., and Bonastre, J.-F., "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4773–4776, IEEE, 2012.

[26] Larcher, A., Bonastre, J.-F., and Mason, J., "Reinforced temporal structure information for embedded utterance-based speaker recognition." in *INTERSPEECH*, pp. 371–374, 2008.

[27] Larcher, A., Lee, K. A., Martınez, P. L. S., Nguyen, T. H., Ma, B., and Li, H., "Extended RSR2015 for text-dependent speaker verification over VHF channel," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[28] Larcher, A., Lee, K. A., Ma, B., and Li, H., "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7673–7677, IEEE, 2013.

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

On the Influence of Text Content on Pass-Phrase Strength for Short-Duration Text-Dependent Automatic Speaker Authentication

Giacomo Valenti[1,2], *Adrien Daniel*[1] *and Nicholas Evans*[2]

[1]NXP Software, Sophia Antipolis, France
[2]EURECOM, Biot, France

giacomo.valenti@nxp.com, adrien.daniel@nxp.com, evans@eurecom.fr

Abstract

In the context of automatic speaker verification it is well known that different speech units offer different levels of speaker discrimination. For short-duration, text-dependent automatic speaker recognition, a user's pass-phrase bears influence on how reliably they can be recognized; just as is the case with text passwords, some spoken pass-phrases are more secure than others. This paper investigates the influence of text or phone content on recognition performance. This work is performed using the shortest duration subset of the standard RSR2015 database. With a thorough statistical analysis, the work shows how significant reductions in error rates can be achieved by preventing the use of weak passwords and that improvements in performance are consistent across disjoint speaker subsets. The ultimate goal of this work is to develop an automated means of enforcing the use of stronger or more discriminant spoken pass-phrases.
Index Terms: speaker recognition, text-dependent, short duration performance evaluation

1. Introduction

The performance of automatic speaker verification (ASV) technology is now sufficient to support mass-market, consumer applications [1]. Most of these, for instance smart phone, smart service applications and those within the sphere of the Internet of Things (IoT), call for short-duration enrolment and recognition, implying text-dependent recognition. While gaining momentum since the release of the RSR2015 [2] and RedDots [3] corpora, research in this area lags behind that in text-independent recognition.

The seminal work in [4] investigated differences in recognition performance at the speaker level, characterising four different speaker classes referred to as Doddington's menagerie. Later work in [5] investigated the influence on performance of specific training utterances. This work aimed to go beyond Doddington's menagerie and to investigate the role of phonetic content on ASV performance. With substantial variation in performance being observed, this raises the question of exactly what speech content is most relevant for speaker discrimination.

The work in [5] was extended in [6] which analysed the idiosyncratic information contained in French vowels. While perhaps offering greater insights relevant to the forensic branch of speaker recognition in terms of explaining results, the work points towards a mechanism for the selection or weighting of the most discriminant speech components for speaker modelling and recognition [7].

Most of the past work detailed above focuses on text-independent recognition where the tradition of speaker recognition evaluation (SRE) campaigns administered by the National Institute of Standards and Technology (NIST) generally dictates relatively long-duration training and testing. When speech data is plentiful, phonetic variation is naturally normalised to some extent. This is not the case for short-duration training and testing where speech data is sparse. In this case, phonetic variation can have a significant impact on recognition performance [8, 9]. Herein lies the contribution of our research.

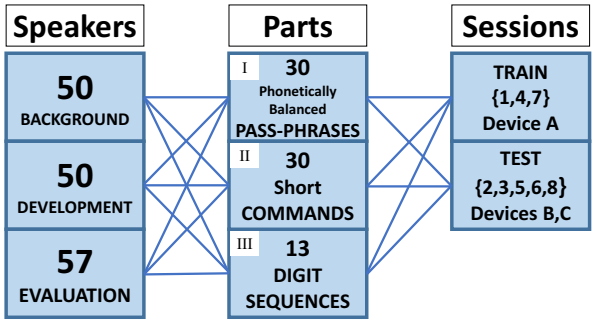


Figure 1: *RSR2015 Database partition for male speakers. The partition is identical for female speakers but with 43 speakers in the evaluation set instead of 57.*

This paper investigates the influence of text content on short-duration, text-dependent speaker recognition. The aim is to assess the variability in recognition performance and to determine the extent to which such variability is consistent across speakers. This work calls for a thorough statistical analysis which is reported here.

The remainder of this paper is organised as follows. Section 2 expands on the motivation for this work and identifies the database and protocols used for it. Section 3 describes the ASV system and results. The statistical analysis of command strength is described in Section 4.

2. Database and protocols

This section describes the database and text-dependent ASV system used for the work reported in this paper.

2.1. Database

The ultimate goal of this work is to develop a system to detect and prevent automatically the use of weak spoken passwords. Such a system would necessarily draw upon the use of speech data collected from other speakers; the only speaker-specific data available at enrolment would be one, or a small number of repetitions of the speaker's chosen password.

Table 1: *The four possible kinds of trials for a text-dependent speaker verification system. They involve different combinations of matching speakers and text.*

Match	Speaker	Text
Target Correct (TC)	Yes	Yes
Target Wrong (TW)	Yes	No
Impostor Correct (IC)	No	Yes
Impostor Wrong (IW)	No	No

As a consequence, weak passwords are thus assumed to be universally weak, that is to say not specific to a given speaker. Required to support this work then, is a corpus collected from different speakers with multiple repetitions of the same set of sentences. The so-called sly impostor subset and associated protocol of the RSR2015 database [10] is ideally suited and is used for all work reported in this paper. The RSR2015 database partition is illustrated in Fig. 1. The sly impostor condition involves matched content impostor trials, sometimes referred to as the impostor-correct (IC) condition. This is one of four possible trials illustrated in Table 1.

The RSR2015 database contains phonetically-balanced sentences (part I), short commands (part II) and random digit trials (part III) (see Fig. 1). Since the target application of this work involves short spoken passwords, all experiments reported here are based upon the short commands condition (part II) where utterances contain in the order of 0.5 seconds of speech.

2.2. Protocols

As illustrated in Fig. 1, there are 50 male and female speakers in the background subset and 50 male and female speakers in the development subset. The evaluation subset is comprised of 57 male speaker and 43 female speakers. Each speaker provides recordings in 9 sessions. Data collected from 3 of the 9 sessions are set aside for training while the remaining 6 are used for testing. When experimenting on Part II, only Part I data is used for the learning of background information and there is no overlap between speakers or phrases between the data used for background modelling and that used for training and testing.

The sly impostor subset of Part II of the RSR2015 corpus contains 8990 TC (target) and 440510 IC (impostor) trials for the development set and 10250 TC and 574000 IC trials for the evaluation set. These numbers differ slightly from those reported in [11][1]. Since the literature focuses on results for the phonetically-balanced pass-phrases of Part I – this is the standard protocol distributed with the RSR2015 database – this paper also reports results for the same standard protocol. The Part I protocol dictates speaker-specific models which are trained with all 30 pass-phrases across the 3 training sessions, giving a total of 90 utterances. Speaker and pass-phrase models are trained with 3 utterances.

3. ASV system

Reported here is the ASV system architecture including details of the modelling and features together with results. While the contribution of this paper is not linked to advances in ASV technology, results are included as a means of illustrating performance relative to the state of the art.

[1]The authors became aware of the standard protocols for RSR 2015 Part II only after most of the work reported here was already completed.

Table 2: *Comparison of results for Part I of the RSR2015 database. Results shown for our implementation of the HiLam system with original results reported in [12]. Results are reported in terms of EER.*

Speaker set	Ours	Larcher et al. [12]
Part I Development	1.74%	1.43%
Part I Evaluation	1.93%	1.33%

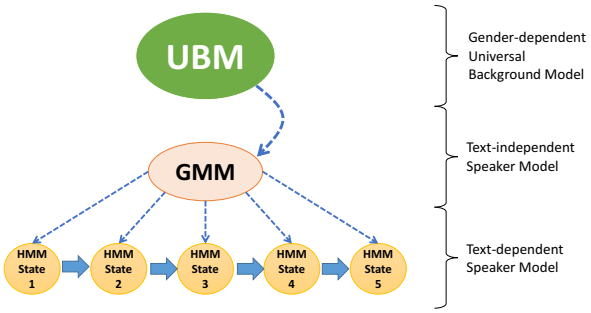


Figure 2: *HiLam system architecture, reproduced from [10].*

3.1. Architecture

The baseline text-dependent ASV system used for all work reported in this paper is our own implementation of the so-called HiLam system originally reported in [11]. As illustrated in Fig. 2, the system is comprised of 3 layers: (i) a gender-dependent universal background model (UBM); (ii) speaker-specific Gaussian mixture models (GMMs) and (iii) speaker-and-text-specific hidden Markov models (HMMs).

The speaker-specific GMM model is obtained from the maximum a posteriori (MAP) adaptation of the UBM. The former is text-independent and does not model any time-sequence information; this is reflected only in the lower text-dependent level. Each HMM state is initialised with the same, second-level GMM model before Viterbi realignment and retraining. The full HiLam training and testing procedures are described in the original work [11].

In our implementation, GMM models have 64 components. MAP adaptation is applied with relevance factors of 19 and 3 for the second and third layers respectively. Scores are conventional log-likelihood ratios calculated between the claimed model and the UBM.

3.2. Feature extraction

The original RSR speech files are pre-processed with silence removal, by calculating the speech active level as recommended in ITU-T P.56 and by thresholding at 15.9 dB. This typically labels in the order of 64% of data for further processing; the remaining high-energy speech data is then frame blocked into 20ms frames with 10ms overlap. Standard MFCC features are then extracted in the usual way. They are comprised of 18 coefficients, without C0, which are appended with deltas and double deltas to produce features of 54 coefficients.

3.3. Performance

Table 2 shows a comparison of ASV results obtained with our implementation of the HiLam system with those reported in the original work [12] for Part I (phonetically-balanced pass-

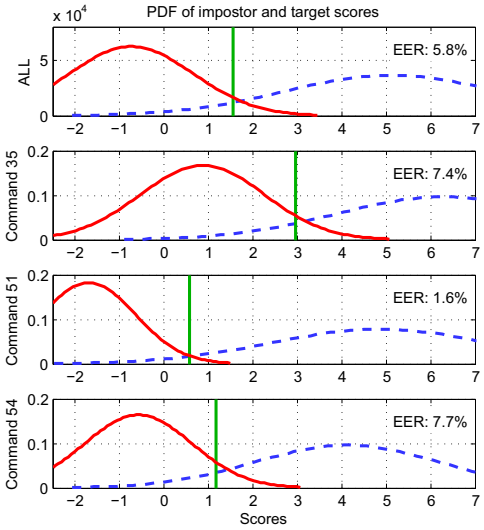


Figure 3: *Impostor (solid red) and target (blue dashed) score distributions and EER thresholds (green vertical lines). Plots illustrated separately for all commands trials (top) and for 3 command-specific trials.*

phrases). Results are reported in terms of EER. All results correspond to the IC condition and show a respectable level of performance; our results are only marginally worse than those reported in [12].

4. Statistical analysis of password strength

Both speaker characteristics and text content influence ASV score distributions. Example target and impostor distributions are illustrated in the top row of Fig. 3. Accept and reject decisions are made according to a *global threshold* illustrated by the vertical green line between the modes of each distribution. The amount of overlap between the two will then determine the *global EER*. The threshold is an inevitable compromise between the 'inner' target and impostor distributions related to an array of different factors, e.g. speaker-dependency, device-dependency and, in this case, text-dependency.

In the case of the IC condition, the influence of text is quantifiable from the target and impostor score distributions for subsets of same-text trials. These distributions are referred to as *text-dependent distributions* and the corresponding distribution overlap as the *text-dependent overlap*. As illustrated in Fig. 3 for commands 35, 51 and 54 of the RSR2015 database there is thus a *text-dependent EER* obtained with a *text-dependent threshold* for each command. The global EER is thus affected by both the text-dependent overlaps and the variation in the text-dependent thresholds. In contrast, text-dependent EERs are affected only by the text-dependent overlaps.

The following sections describe a statistical analysis that illustrates the potential to improve ASV performance through the selection of strong spoken sentences. It furthermore demonstrates that the notion of password strength is consistent across disjoint sets of speakers.

4.1. Variable strength command groups

The following describes a process to rank commands in terms of strength. This is needed in order to simulate a text-dependent ASV system that would eventually include password strength recommendation. On the assumption that a strong password is characterised by a relatively small text-dependent overlap, commands are first ranked by decreasing text-dependent EER. This process is performed separately for the development and evaluation sets thus yielding two rankings. From each of these rankings, groups of commands are formed by selecting 10 with the closest strength starting at every rank position, thereby producing 21 groups in total. The first group is comprised of the 10 weakest commands ranked #1 to #10, the second group is comprised of those ranked #2 to #11 and so on until the last group which contains the 10 strongest commands ranked #20 to #30. It is stressed that, while the groups obtained for the development and evaluation sets are similar, they are not identical.

4.2. Sampling distribution of the EER

ASV performance is assessed independently for each group in terms of the global EER (encompassing all commands in each group). The significance of the difference in recognition performance obtained for each group is measured with the following bootstrapping procedure [13].

For each group, a thousand populations of 30 commands are generated by picking at random from the 10 commands in the group. This procedure is known as resampling with replacement [13]. Each resampling of 30 commands out of 10 produces a population whose size is the same as that of the full dataset in terms of the number of trials. Each of these sampled populations yields an EER value which is computed from the target and impostor trials of the 30 commands of the population. These 1000 EERs form a sampling distribution of the global EER for each group.

The sampling distributions were visually inspected for normality, allowing for 95% confidence intervals of 1.96 times the standard deviation of the distribution, thereby removing 2.5% of the observations at each end of the distribution. This interval around the mean EER of the distribution has a high probability of encompassing the true value of the EER for each group. Differences in performance obtained for groups with non-overlapping confidence intervals can hence be considered as being statistically significant.

The bootstrapping procedure is applied using four combinations of different ranking and trial sets: (i) ranking and trials both for the development set, (ii) ranking and trials both for the evaluation set, (iii) ranking for the development set and trials for the evaluation set, and (iv) ranking for the evaluation set and trials for the development set. Combinations (iii) and (iv) are necessary in order to illustrate whether or not command strength is consistent across disjoint speaker sets. Statistics obtained for combinations (i) and (ii) are depicted in Fig. 4(a) and 4(b) by solid symbols in each plot. Statistics obtained for combinations (iii) and (iv) are depicted by unfilled symbols.

4.3. Isolating the influence of overlap

ASV performance estimated for each group is the consequence of the variation in text-dependent overlaps and text-dependent thresholds in each group. To illustrate the dependence on overlap in isolation from threshold effects, the experiments described above are repeated with all trial scores normalised according to the text-dependent threshold. The text-dependent EER for each command is then obtained with a score threshold of zero. Results for this experiment are reported in Fig. 4(c) and (d).

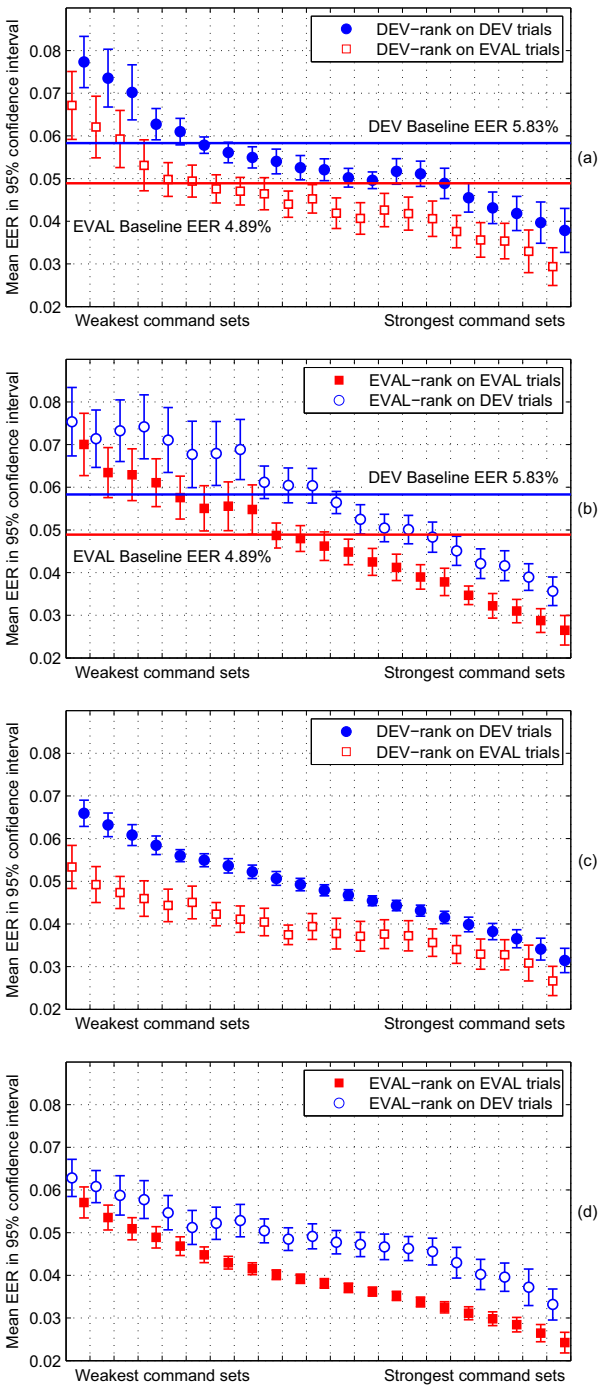


Figure 4: *ASV performance with (c,d) and without (a,b) text-dependent threshold adjustment. Each point represents the mean EER over 1000 resamplings of 30 commands chosen with replacement among the 10 commands of each sub group. The horizontal lines in (a,b) represent the baseline performance of the system for both sets with all 30 commands.*

4.4. Results interpretation

When using their own ranking, EER results for both the development and evaluation sets show significant decreases as the group contains increasingly stronger commands – solid-symbol plots in Fig. 4(a) and 4(b). When using threshold-adjusted scores (solid-symbol plots in Fig. 4(c) and 4(d), decreases are strictly monotonic. This observation confirms that the spread of text-dependent thresholds also affects performance.

Other observations concern results for cross-set rankings – unfilled-symbol plots in Fig. 4(c) and 4(d). Rankings made on the development set translate well to the evaluation set and vice-versa. For the evaluation set, results illustrated in Fig. 4(a) show that only 6 groups have an EER which is not significantly different to the overall EER (4.89%). For the development set, results illustrated in Fig. 4(b) show only 4 groups with a non-significantly different overall EER (5.83%). The significant global decrease in EER (albeit non-monotonic) shows that, with negligible differences in ranking, some commands are consistently 'weak' across different speakers. According to these results, a system including a password strength acceptance criterion could halve the error rate by choosing stronger sentences over weaker ones (from 5.34% to 2.67% on the development set, and from 6.28% to 3.32% on the evaluation set). Finally, we note that the visible offset of the evaluation set EERs is inherent to the RSR2015 database and consistent with results presented by others [11, 12].

The factors responsible for the ranking of command strength are not addressed in this paper, thus a solution to identify automatically weak short sentences is left for future work. Some intuitive, high-level observations are nonetheless offered. Consistent to both development and evaluation sets is the higher ranking of longer duration sentences. This is not surprising. Other observations are more intriguing. While commands such as 'Turn on light', 'Watch Cartoon' and 'Volume Down', all of similar duration, all perform well across both subsets, others of similar length such as 'Door Open', 'Volume up' and 'Aircon off' performed poorly across both subsets. Given the similar duration, it is assumed that the first three commands have more discriminative phonetic content. 'Volume up' and 'Volume down' vary only by the last two phonemes but are ranked among the weakest and strongest commands respectively. These observations are consistent with the discriminative power of nasal sounds studied in [7]. Clearly these factors warrant further attention in future work.

5. Conclusions and future work

This paper investigates short-duration, text-dependent automatic speaker authentication. The contribution relates to a thorough statistical analysis of the influence of text content on command strength. This not only influences the optimum system threshold, but also the degree of overlap between target and impostor score distributions. As a result, some spoken commands are stronger than others.

In order to examine the impact of text on the overlap between target and impostor score distributions and hence ASV performance, the influence of the threshold is compensated for *a posteriori*. Automatic means to compensate or normalise for this influence is an issue for future work. The ranking of commands according to their strength reveals considerable differences in their impact on system performance. The next stage of this work is to develop an automatic means of identifying weaker spoken short sentences. The intention is to develop such a system for a real-use case scenario in which the user of an ASV system may be encouraged to use a *strong* spoken sentence, namely one which offers a high level of discrimination among different speakers.

6. References

[1] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," IEEE SLTC Newsletter, February 2013.

[2] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "RSR2015: Database for text-dependent speaker verification using multiple pass-phrases." in *INTERSPEECH*, 2012, pp. 1580–1583.

[3] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brummer, D. van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, H. Li, T. Stafylakis, J. Alam, A. Swart, and J. Perez, "The RedDots data collection for speaker recognition," in *INTERSPEECH*, 2015, pp. 2996–3000.

[4] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation," in *DTIC Document*, 1998.

[5] J. Kahn, S. Rossato, and J.-F. Bonastre, "Beyond doddington menagerie, a first step towards," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 4534–4537.

[6] J. Kahn, N. Audibert, J.-F. Bonastre, and S. Rossato, "Inter and intraspeaker variability in french: an analysis of oral vowels and its implication for automatic speaker verification," in *International Congress of Phonetic Sciences (ICPhS)*, 2011, pp. 1002–1005.

[7] K. Amino, T. Sugawara, and T. Arai, "Idiosyncrasy of nasal sounds in human speaker identification and their acoustic properties," in *Acoustical science and technology*, vol. 27, no. 4, 2006, pp. 233–235.

[8] B. Fauve, N. Evans, and J.Mason, "Improving the performance of text-independent short duration SVM-and GMM-based speaker verification," in *Odyssey*, 2008, p. 18.

[9] G. Soldi, S. Bozonnet, F. Alegre, C. Beaugeant, and N. Evans, "Short-duration speaker modelling with phone adaptive training," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[10] A. Larcher, K.-A. Lee, B. Ma, , and H. Li, "RSR2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Interspeech 2012*.

[11] A.Larcher, K. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[12] A. Larcher, K. A. Lee, P. L. S. Martnez, T. H. Nguyen, B. Ma, and H. Li, "Extended RSR2015 for text-dependent speaker verification over VHF channel," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[13] D. Howell, *Statistical methods for psychology, 7th edition*. Cengage Learning, 2010.

US 20180060557A1 115

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2018/0060557 A1**

Valenti et al. (43) **Pub. Date: Mar. 1, 2018**

(54) **SPOKEN PASS-PHRASE SUITABILITY DETERMINATION**

(71) Applicant: **NXP USA, Inc.**, Austin, TX (US)

(72) Inventors: **Giacomo Valenti**, Antibes Juan-les-Pins (FR); **Adrien Daniels**, Antibes (FR); **Nicholas Evans**, Valbonne (FR)

(21) Appl. No.: **15/685,146**

(22) Filed: **Aug. 24, 2017**

(30) **Foreign Application Priority Data**

Aug. 25, 2016 (EP) 16290162.3

Publication Classification

(51) **Int. Cl.**

G06F 21/32	(2006.01)
G10L 15/14	(2006.01)
G10L 15/16	(2006.01)
G10L 15/187	(2006.01)
G10L 17/24	(2006.01)
G06F 21/46	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 21/32*** (2013.01); ***G10L 15/142*** (2013.01); ***G06F 21/46*** (2013.01); ***G10L 15/187*** (2013.01); ***G10L 17/24*** (2013.01); ***G10L 15/16*** (2013.01)

(57) **ABSTRACT**

An apparatus comprising at least one processor and at least one memory including computer program code,

the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

provide for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

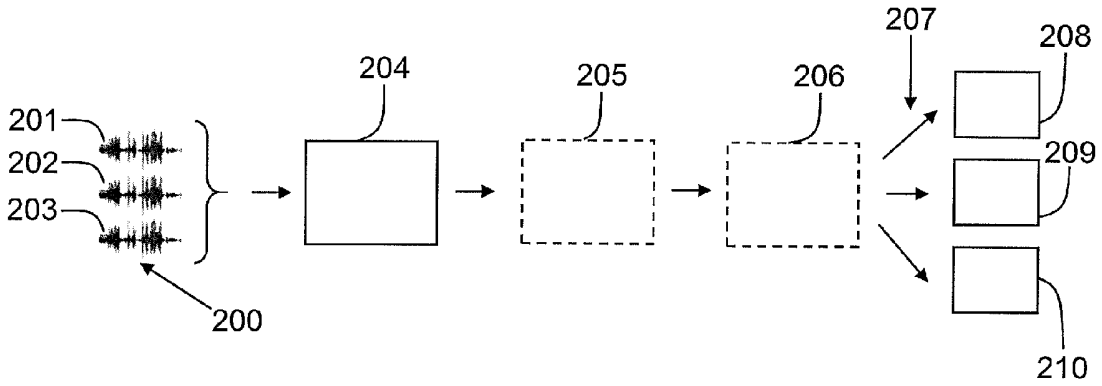


Figure 1

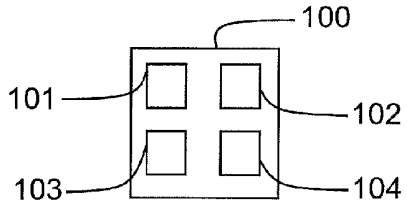


Figure 2

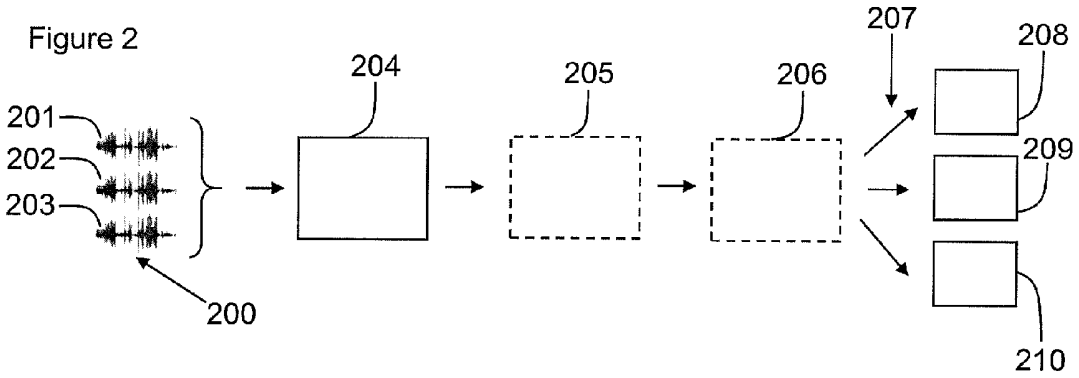


Figure 3

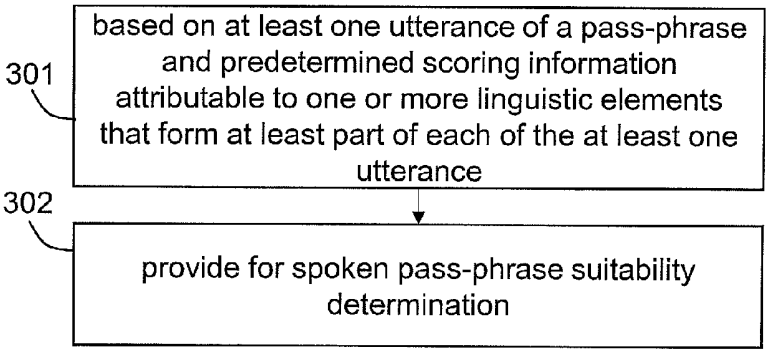
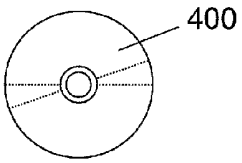


Figure 4



SPOKEN PASS-PHRASE SUITABILITY DETERMINATION

[0001] The present disclosure relates to an apparatus for determining the suitability of a spoken pass-phrase. It also relates to an associated method and computer program for determining the same.

[0002] It may be desirable for a spoken pass-phrase to be strong, similar to a text based password.

[0003] According to a first aspect of the present disclosure there is provided an apparatus comprising at least one processor and at least one memory including computer program code,

[0004] the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

[0005] based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

[0006] provide for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

[0007] In one or more embodiments, the apparatus provides for control of a spoken pass-phrase enrolment procedure for providing a spoken pass-phrase for future authentication of a user, based on the suitability determination.

[0008] In one or more embodiments, the apparatus is configured to base said spoken pass-phrase suitability on at least two utterances of the pass-phrase and the pass-phrase suitability is also determined to be deficient based on a measure of spoken pass-phrase consistency comprising a difference between the at least two utterances being above a predetermined consistency threshold.

[0009] In one or more embodiments, the linguistic analysis comprises one or more of:

[0010] phonetic analysis, glottal voice source feature analysis, morpheme analysis, prosodic unit analysis, phonological analysis, syllable analysis, onset and rime analysis, articulatory features analysis and mora analysis.

[0011] In one or more embodiments, the linguistic elements comprise one or more of:

[0012] words, syllables, phonetic parts, prosodic units, patterns of two or more phonetic parts in the at least two utterances, patterns of two or more prosodic units in the at least two utterances.

[0013] In one or more embodiments, on determination that the pass-phrase suitability is deficient based on the pass-phrase score being below the predetermined pass-phrase score threshold, the apparatus is configured to provide for prompting of a user to change their pass-phrase.

[0014] In one or more embodiments, on determination that the pass-phrase suitability is deficient based on the difference between the at least two utterances being above a predetermined consistency threshold, the apparatus is configured to provide for prompting of a user to make one or more further utterances of the pass-phrase.

[0015] In one or more embodiments, the apparatus is configured to generate a pass-phrase model from the at least two utterances of the pass-phrase, the pass-phrase model comprising a statistical description of the utterances of the pass-phrase, wherein the measure of spoken pass-phrase consistency comprises a difference between the model and a corresponding statistical description of at least one of the at least two utterances.

[0016] In one or more embodiments, the measure of spoken pass-phrase consistency comprises a log-likelihood ratio.

[0017] In one or more embodiments, the pass-phrase model comprises one or more of: a hidden Markov based model, a Gaussian mixture model, i-Vector probabilistic linear discriminant analysis model and a neural network based model.

[0018] In one or more embodiments, the linguistic elements comprises phonemes and/or phones. In one or more embodiments, the apparatus is configured to provide for segmentation of the one, or each, utterance of the pass-phrase into a plurality of individual phonemes and/or phones, and wherein each of the plurality of phonemes and/or phones is assigned its linguistic-element-score in accordance with the predetermined scoring information, the pass-phrase score based on the linguistic-element-scores for the plurality of phonemes and/or phones.

[0019] In one or more embodiments, the pass-phrase suitability is also determined to be deficient based on an identification of insufficient linguistic elements that have a linguistic-element-score above a distinctiveness threshold using a minimum distinctiveness threshold.

[0020] In one or more embodiments, the apparatus is configured to provide for the spoken pass-phrase suitability determination as part of an enrolment procedure in which a user provides a spoken pass-phrase for future use to authenticate the identity of the user.

[0021] In one or more embodiments, the apparatus comprises at least part of: portable electronic device, a mobile phone, a Smartphone, a laptop computer, a desktop computer, a tablet computer, a personal digital assistant, a digital camera, a smartwatch, a non-portable electronic device, a monitor, a household appliance, a smart TV, a server, or a module/circuitry for one or more of the same.

[0022] According to a second aspect of the present disclosure there is provided a method comprising;

[0023] based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

[0024] providing for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on

the pass-phrase score being below a predetermined pass-phrase score threshold.

[0025] According to a third aspect of the present disclosure there is provided a computer readable medium comprising computer program code stored thereon, the computer readable medium and computer program code being configured to, when run on at least one processor having memory, perform at least the following:

[0026] based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

[0027] providing for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

[0028] While the disclosure is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that other embodiments, beyond the particular embodiments described, are possible as well. All modifications, equivalents, and alternative embodiments falling within the spirit and scope of the appended claims are covered as well.

[0029] The above discussion is not intended to represent every example embodiment or every implementation within the scope of the current or future Claim sets. The figures and Detailed Description that follow also exemplify various example embodiments. Various example embodiments may be more completely understood in consideration of the following Detailed Description in connection with the accompanying Drawings.

[0030] One or more embodiments will now be described by way of example only with reference to the accompanying drawings in which:

[0031] FIG. **1** shows an example embodiment of an apparatus;

[0032] FIG. **2** shows an illustration of the functional parts of the apparatus;

[0033] FIG. **3** shows a flowchart illustrating a method; and

[0034] FIG. **4** illustrates a computer readable medium.

[0035] FIG. **1** shows an apparatus **100** comprising memory **101**, a processor **102**, input **103** and output **104**. In this embodiment only one processor and one memory are shown but it will be appreciated that other embodiments may utilise more than one processor and/or more than one memory (e.g. same or different processor/memory types).

[0036] In this embodiment the apparatus **100** is an Application Specific Integrated Circuit (ASIC) for a voice controlled (or at least voice authenticated) electronic device. In other embodiments the apparatus **100** can be a module for such a device, or may be the device itself, wherein the processor **102** may be a general purpose CPU of the device and the memory **101** may be general purpose memory comprised by the device. In other embodiments, the apparatus may be part of a voice control or voice authentication or voice authentication enrolment device. The functionality of the apparatus may be distributed over a plurality of processing devices, which may be remote from one another and in communication via a network or the like.

[0037] The input **103** allows for receipt of signalling to the apparatus **100** from further components, such as a microphone for receipt of one or more utterances or one or more other components configured to provide for pre-processing of utterances detected by a microphone and/or an input device such as a touch-sensitive or the like. The output **104** allows for onward provision of signalling from within the apparatus **100** to further components such as a display screen, a speaker or other components that require confirmation that a spoken pass-phrase is suitable for future use. In this embodiment the input **103** and output **104** are part of a connection bus that allows for connection of the apparatus **100** to further components.

[0038] The processor **102** is a processor dedicated to executing/processing information received via the input **103** in accordance with instructions stored in the form of computer program code on the memory **101**. The output signalling generated by such operations from the processor **102** is provided onwards to further components via the output **104**.

[0039] The memory **101** (not necessarily a single memory unit) is a computer readable medium (such as solid state memory, a hard drive, ROM, RAM, Flash or the like) that stores computer program code. This computer program code stores instructions that are executable by the processor **102**, when the program code is run on the processor **102**. The internal connections between the memory **101** and the processor **102** can be understood, in one or more example embodiments, to provide an active coupling between the processor **102** and the memory **101** to allow the processor **102** to access the computer program code stored on the memory **101**.

[0040] In this example the input **103**, output **104**, processor **102** and memory **101** are all electrically connected to one another internally so as to be integrated together as a single chip that can be installed into an electronic device. In other examples one or more or all of the components may be located separately from one another.

[0041] The apparatus **100** is configured to provide for spoken pass-phrase suitability determination. The apparatus may be considered to determine the suitability based on an assessment of pass-phrase strength. The spoken pass-phrase may be user determined as opposed to a user saying a standard word or phrase provided to them and therefore determining whether or not the user's chosen spoken pass-phrase is suitable for control or authentication may be advantageous. In one or more examples, the apparatus **100** is configured to provide for biometric assessment of the spoken pass-phrase in determination of its suitability. In one or more examples, the apparatus **100** may be configured to be used as part of an enrolment procedure in which a user provides a spoken, user-determined, pass-phrase that forms the basis for future authentication of that user to access or operate a particular device or service. Thus, the apparatus **100** may be used for a telephone or online banking system during enrolment of the user to use that system in the future. In one or more examples, the apparatus **100** may be used by a smart TV to set up a spoken pass-phrase for authenticated access to particular settings, such as parental control settings. The apparatus **100** may be configured to provide

feedback to a user who has provided a spoken pass-phrase on whether or not said spoken pass-phrase is suitable for future access or control. The suitability of a spoken pass-phrase may be based on one or more of: a measure of the distinctiveness of pass-phrase (such as how likely the same pass-phrase would be spoken or pronounced the same way by someone other than the user, which may be considered a biometric check) and a measure of the consistency with which the user is able to articulate or pronounce the same pass-phrase (such as to ensure the chosen pass-phrase is reliably repeatable).

[0042] FIG. **2** shows diagrammatically the functional elements of the apparatus **100**. In particular, FIG. **2** shows the receipt **200** of at least one utterance of the same pass-phrase. In particular, in this example, the apparatus **100** is configured to receive three utterances of the same pass-phrase shown as **201**, **202** and **203**. It will be appreciated that the number of utterances may vary and may depend on how determination of whether or not the pass-phrase is suitable is implemented. For example, the apparatus may determine that the phrase is unsuitable based on the pass-phrase meeting any one of one or more criteria, as will be described below. Some of the criteria may require different numbers of utterances on which to determine the suitability.

[0043] The apparatus **100** may be configured to provide for analysis of the or each utterance by analysis of spoken linguistic features comprising features of language or component parts of language, such as phones or phonetics. A linguistic analysis component **204** provides for linguistic analysis of each of the three utterances **201-203**. Accordingly, the determination of the suitability of the pass-phrase may be based on one or more linguistic characteristics thereof. The linguistic analysis component **204** may be provided with predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements. Thus, as the pass-phrase spoken in the utterances **201-203** is user-selected, the apparatus **100** may not be aware of which linguistic elements may be present. Accordingly, the predetermined scoring information may comprise a library of linguistic elements that may potentially be present in the spoken pass-phrase utterances **201-203**. Nevertheless, the linguistic analysis may require that the predetermined scoring information contains linguistic elements that form at least part of the utterances **201-203**. The predetermined scoring information may be stored in the memory **101** or may be stored remotely and accessed via a communication element (not shown).

[0044] The linguistic analysis component **204** may be configured to assign a pass-phrase-score to each utterance of the pass-phrase based on linguistic analysis. Accordingly, the linguistic analysis component may be configured to identify one or more linguistic elements in said utterances **201-203**. With reference to the predetermined scoring information, the or each linguistic element is given a linguistic-element-score by, for example, look up of the linguistic element identified in the spoken pass-phrase in the predetermined scoring information. Certain linguistic elements may be given higher scores by the predetermined scoring information because their presence in the spoken pass-phrase is distinctive or the way the particular user says them is distinctive. Other linguistic elements may be given a lower score perhaps because they are undistinctive or the way the particular user says them is undistinctive.

[0045] The linguistic analysis component **204** may be configured to determine the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements. For example, the pass-phrase score may be sum of the linguistic-element scores, a product of the linguistic element scores, a function of the linguistic element scores. For example, the pass-phrase score may be determined, at least in part, based on the number, pattern, and/or frequency of occurrence of one or more of the linguistic elements, different linguistic elements, linguistic elements that have a particular linguistic element score such as above a distinctiveness threshold, a group of two or more temporally adjacent linguistic elements, and a group of two or more linguistic elements appearing within a predetermined temporal distance of one another in the spoken pass-phrase.

[0046] For example, it may be provided for in the predetermined scoring information, that if a pattern of consecutive (or regular) phonemes and/or phones of a particular family of phonemes and/or phones is identified, and such a pattern is known to be very discriminative, then a high linguistic element score may be applied. In one or more examples, if the linguistic analysis component identified a certain prosody (relative variation of fundamental frequency and/or derivatives) and such a prosody is known to be very discriminative (maybe regardless of the component, non-discriminative phonemes that form it) then a high linguistic element score may be applied.

[0047] One or more linguistic elements that have a linguistic-element score below a particular distinctiveness threshold may be excluded from determination of the pass-phrase score. This may be advantageous as a spoken-pass-phrase having many low scoring linguistic elements may not be, overall, very distinctive. However, a shorter pass-phrase having a few high scoring linguistic elements may be more distinctive overall. In one or more examples, patterns of linguistic elements may be identified prior to setting the distinctiveness threshold, because it may be found that a particular pattern of undiscriminating linguistic element is, itself, discriminative of the user.

[0048] FIG. **2** further shows an optional model determination component **205** and an optional model analysis component **206**, which will be described below.

[0049] The apparatus **100** is configured to provide for spoken pass-phrase suitability determination by assessment of the pass-phrase score at **207**. The spoken pass-phrase suitability may be determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold. This determination of unsuitability is shown as a first deficient pass-phrase action component **208**. The first deficient pass-phrase action component **208** may provide for feedback to the user or another system (such as a system enrolling the user for a service) that the pass-phrase spoken by the user is unsuitable, as there may be a (too) high likelihood of the pass-phrase being spoken or pronounced the same way by someone else. The feedback may comprise a prompt to the user to choose a different pass-phrase.

[0050] The spoken pass-phrase suitability may be determined to be acceptable at least based on the pass-phrase score being equal to or above the predetermined pass-phrase score threshold. This determination of suitability is shown as acceptable pass-phrase action component **209**. The component **209** may provide for feedback to the user or another system (such as a system enrolling the user for a service) that

the pass-phrase spoken by the user is suitable. In one or more examples, the enrolment procedure may be continued or completed based on this determination.

[0051] The utterance of a pass-phrase may comprise at least one or more spoken words or syllables. In one or more examples, the utterance comprises at least two, three, four or five spoken words or syllables. In one or more examples, the duration of the utterance comprises at least 0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.5 or at least 2.0 seconds. In one or more examples, the duration of the utterance comprises less than 10, 8, 6, 5, 4, 3 seconds.

[0052] The predetermined scoring information may be based on analysis of a large corpus of spoken words. The analysis may identify linguistic elements that have the widest range of pronunciation amongst a given set of speakers. Scores may be assigned empirically or based on a predetermined function to each linguistic element appearing in the predetermined scoring information. The predetermined scoring information may comprise a look-up table of linguistic elements and associated linguistic-element scores. The predetermined scoring information may comprise a function to assign linguistic elements an associated linguistic-element score based on one or more of: their presence in the utterance, their existence with other linguistic elements in the utterance and their existence immediately adjacent to other linguistic elements in the utterance (or within a temporal/linguistic element distance). The scoring of the linguistic elements compared to the counting of the quantity of linguistic elements may be advantageous, as the suitability of a pass-phrase may, in some examples, be better judged by the quality of its linguistic content rather than the quantity.

[0053] In one or more examples, the linguistic analysis may comprise phonetic analysis. Accordingly, the linguistic elements comprises phonemes and/or phones and the apparatus is configured to provide for segmentation of the one, or each, utterance of the pass-phrase into a plurality of individual phonemes or phones or groups of phonemes or phones. Those skilled in the art will recognize that techniques are known for parsing an utterance into phonemes. For example, J Manikandan et al "*Implementation of a Novel Phoneme Recognition System Using TMS320C6713 DSP,*" 2010, pp. 27-32 provides an example of such a technique. It will be appreciated that other phoneme parsing/labelling techniques may be used. In this example, the predetermined scoring information comprises scoring for phonemes. For example, it is known from prior studies of speakers that particular phonemes are more distinctive than others, such as nasal vowels. Accordingly, the scores provided by the predetermined scoring information may be based on such studies. Each of the phonemes identified in the utterance can be assigned its linguistic-element-score in accordance with the predetermined scoring information. The utterance of the pass-phrase may then be provided with a pass-phrase score based on the scores given to the phonemes present therein. For example, the pass-phrase score may comprise a sum of the linguistic-element-scores for the plurality of phonemes, although other methods of combining the linguistic-element-scores may be used.

[0054] In one or more examples, the phonetic analysis may include modelling tri-phones and/or bi-phones such as by using hidden Markov models and the scoring is applied to the modelled linguistic (phonetic) elements.

[0055] The linguistic analysis component **204** may provide for splitting the utterance into a plurality of frames (i.e. temporal sub-sections), which may be overlapping (in time). The identification of phonemes, phones or other linguistic elements may be applied to the frames. This may provide for more efficient processing.

[0056] In one or more examples, as described above, the linguistic analysis may comprise phonetic analysis, but in one or more examples, other forms of linguistic analysis may be performed. For example, the method may comprise identifying prosodic units in the utterance and the assignment of linguistic-element scores to each of the identified prosodic units perhaps using the look-up table or function technique above. The prosodic analysis may be provided in terms of f_0 dynamics.

[0057] In one or more examples, and as described above, the linguistic analysis may comprise one or more of phonetic analysis, glottal voice source feature analysis, morpheme analysis, prosodic unit analysis, phonological analysis, syllable analysis, onset and rime analysis, articulatory features analysis and/or mora analysis. Accordingly, the linguistic element may comprise the subject of each type of analysis. One or more types of linguistic analysis may be performed and the pass-phrase score may be a combination of different linguistic-element scores based on the different types of linguistic analysis.

[0058] For glottal voice source feature analysis, the apparatus may be configured identify the occurrence of and/or patterns in and/or form of glottal sounds in the one or more utterances. The use of glottal analysis may be advantageous as it has been found that glottal sounds may contain information about the state of the vocal tract, which may be useful for an anti-spoofing check.

[0059] In addition to any of the above linguistic analysis techniques or combinations of linguistic analysis, the apparatus may or may not be configured to base the determination of the suitability of the utterance of the pass-phrase on audio features of the utterances. Accordingly, changes in or rates of change in amplitude or frequency or changes in rates of change may be determined and used together with the linguistic analysis to score the pass-phrase. In one or more examples, the audio features may include one or more of frequency fading, cut-off (in terms of time or frequency), static noise cause by deterioration of the microphone and/or ambient noise) or something suspect (for example, every component has the same phase, which may be indicative of the utterance being generated by a voice synthesiser, such as used in a spoofing attack.

[0060] In one or more examples, the linguistic analysis may be applied to the whole utterance rather than identifying sub-elements thereof. Accordingly, the linguistic element may comprises the whole pass-phrase uttered by the user. The predetermined scoring information may provide information for scoring the utterance, such as based on patterns (e.g. linguistic patterns) identified therein rather than individual component parts.

[0061] The linguistic analysis component **204** may be configured to identify patterns in one or more audio features of one or more of the linguistic elements. The audio features may comprise one or more of: volume, tone or changes therein. For example, an audio feature of an identified linguistic feature may be identified that better characterizes how the user said that phoneme. For example, pitch variation or the formant distances or the fact that said phoneme, with particular spectral components, occurs after a sequences of nasal and vowels phonemes and/or at a par-

ticular volume and/or held for at least a particular time may be a good discriminator. The same may be applied for phones rather than or in additional to the phonemes described above.

[0062] In one or more examples, where the apparatus **100** is configured to base said spoken pass-phrase suitability on at least two utterances of the pass-phrase, the suitability of the spoken pass-phrase may also be judged based on how consistently the user spoke the same pass-phrase during the at least two utterances. Thus, the failure to reach a pass-phrase score threshold may comprise a first criteria for declaring the pass-phrase unsuitable and the consistency across utterances of the spoken pass-phrase may be a second criteria. It will be appreciated that the reason for a difference in pronunciation between two or more utterances **201**, **202**, **203** of the same pass-phrase may be due to a mistake by the user, or background noise or because the chosen pass-phrase is simply difficult for that user to articulate consistently. Accordingly, the failure of this consistency criteria may cause the apparatus **100** to provide for feedback, as diagrammatically shown as inconsistent utterance action component **210**. The feedback to the user may be an instruction to them to provide one or more further utterances of the same pass-phase. The consistency may be re-evaluated based on the one or more additional utterances. If the utterances are still not consistent, this may cause the apparatus **100** to provide for feedback to the user instructing them to choose a different pass-phrase and provide utterances of that different pass-phrase. The feedback provided by the inconsistent utterance action component **210** may be to a further device, such as one performing a user enrolment procedure.

[0063] The apparatus **100** may provide for measuring of a spoken pass-phrase consistency of the utterances. To evaluate the spoken pass-phrase consistency a difference between at least two utterances **201**, **202** or **202**, **203** or **201**, **203** may be compared to a predetermined consistency threshold. A correlation function may be used to determine the difference such as the determination of the correlation coefficient. In one or more examples, the utterances may be processed to provide for one or more of removal of any silence or noise at the beginning and/or end and/or during the utterance; noise removal; non-voice noise removal; scaling of the duration within bounds; and aligning the at least two utterances in time. A comparison between the utterances may then be made. This processing may be performed independent of the spoken pass-phrase consistency evaluation, such as prior to assignment of linguistic-element scores.

[0064] In one or more examples, a different method for evaluating consistency may be used, such as based on the consistent identification of the same linguistic elements (e.g. phonemes) at the same temporal locations in the utterance (or within a threshold distance of the same temporal location).

[0065] The evaluation of the consistency of the utterances may be based on a comparison between one of the utterances and a model comprising a statistical description of both or all the utterances.

[0066] The model determination component **205** provides for generation of a model that characterizes the plurality of utterances (such as in combination or individually) in terms of a statistical description of the linguistic content. Thus, in one or more examples, the statistical description may define the utterance in terms of temporal positioning of phonetic or prosodic content or patterns thereof. As shown in FIG. **2**, the model may include one or more parts of the data representing audio of the utterances **201**, **202**, **203**.

[0067] The model may be a hidden Markov model or a Gaussian mixture model of the utterances **201**, **202**, **203**. The model may be generated using a maximum a posteriori (MAP) estimation technique, although other model generation techniques may be used, such as like i-Vector probabilistic linear discriminant analysis (PLDA). The input to the model determination component **205** may comprise the utterances **201**, **202**, **203** or processed versions thereof (as mentioned above and in terms of framed sub-sections). Further, the results of the linguistic analysis component **204** may provide for generation of the model. Thus, the labelling of the linguistic elements in the utterance and/or the score applied to the linguistic elements may be provided for generation of the model. The modelling of utterances using hidden Markov model or a Gaussian mixture model will be familiar to those skilled in the art.

[0068] The linguistic analysis component **204** may provide for pruning (i.e. removal) of linguistic components that score below a minimum distinctiveness threshold. Thus, linguistic components that do not contribute (much) to the distinctiveness of the utterance may be removed from further consideration. Thus, in one or more examples, the input to the model determination component **205** may comprise the utterance with indistinctive linguistic content removed therefrom.

[0069] The suitability of the pass-phrase may further by evaluated based on one or more of the temporal length or a function of the temporal length and scores applied to remaining linguistic elements once the linguistic components that score below a minimum distinctiveness threshold are removed. Thus, the apparatus **100** may require the "pruned" utterance to be above a minimum length of, for example, 1 second. In one or more examples, the apparatus may define different temporal lengths based on the linguistic element scores. Thus, for example, a high scoring spoken pass-phrase may be allowed to be shorter in temporal length than a lower scoring spoken pass-phrase. If this process determines the pass phrase not to be suitable, the apparatus may provide for the feedback provided by deficient pass-phrase action component **208**.

[0070] The model created by the model determination component **205** may, if the pass-phrase is determined to be suitable by the apparatus **100**, forms the description of the utterance of the pass-phrase upon which future utterances of the pass-phrase are compared against to authenticate the user or provide for particular control of device.

[0071] The optional model analysis component **206** may provide for analysis of the suitability of the pass-phrase. As mentioned above, the evaluation of the consistency of the utterances may be based on a comparison between one of the utterances and the model generated by the model determination component **205**. In one or more examples, the model is created from a combination of the at least two utterances (in this example three utterances **201**, **202**, **203**). In the evaluation of consistency, the apparatus **100** takes one or each of the utterances and compares a statistical description of the individual utterance with the (same) statistical description used by the model. The comparison may comprise determination of a log-likelihood ratio which is compared to the consistency threshold. In this case the log-likelihood ratio will be between the log-likelihood of the

generated model and a Universal Background Model (UBM). The UBM comprises a model representative of speaker-independent spoken feature characteristics. The UBM may be gender dependent. N scores may be obtained that correspond to the N pass phrase utterances: in each case, two terms of the log-likelihood ratio depend on the distance between the utterance (its feature vector) and the generated model from the utterances versus the distance between the utterance (its feature vector) and the UBM. Then, the distance between those N scores gives a measure of compactness of the model.

[0072] In the above examples, the apparatus **100** is described as performing the functionality described. However, it will be appreciated that the apparatus may provide, through signalling sent to other components, for the functionality to be performed by those other components. Accordingly, the apparatus **100** may manage the functionality provided by other components or may perform that functionality itself or a combination of the two. For example, the apparatus may provide signalling to a linguistic element identifier in order to identify linguistic elements. The linguistic element identifier may provide the results back to the apparatus. For example, the apparatus may provide signalling to a linguistic element scorer for scoring the linguistic elements, the results being fed back to the apparatus **100**. For example, the apparatus may provide signalling to a model generator for performing the function of the model determination component **205**, the results being fed back to the apparatus **100**. For example, the apparatus may provide signalling to a model analyser for performing the function of the model analysis component **206**, the results being fed back to the apparatus **100**.

[0073] FIG. **3** shows a flow chart illustrating the steps of based on **301** at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance; providing for **302** spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

[0074] FIG. **4** illustrates schematically a computer/processor readable medium **400** providing a program according to an example. In this example, the computer/processor readable medium is a disc such as a digital versatile disc (DVD) or a compact disc (CD). In other examples, the computer readable medium may be any medium that has been programmed in such a way as to carry out a defined function. The computer program code may be distributed between multiple memories of the same type, or multiple memories of a different type, such as ROM, RAM, flash, hard disk, solid state, etc.

[0075] The instructions and/or flowchart steps in the above figures can be executed in any order, unless a specific order is explicitly stated. Also, those skilled in the art will recognize that while one example set of instructions/method has been discussed, the material in this specification can be combined in a variety of ways to yield other examples as well, and are to be understood within a context provided by this detailed description.

[0076] In some example embodiments the set of instructions/method steps described above are implemented as functional and software instructions embodied as a set of executable instructions which are effected on a computer or machine which is programmed with and controlled by said executable instructions. Such instructions are loaded for execution on a processor (such as one or more CPUs). The term processor includes microprocessors, microcontrollers, processor modules or subsystems (including one or more microprocessors or microcontrollers), or other control or computing devices. A processor can refer to a single component or to plural components.

[0077] In other examples, the set of instructions/methods illustrated herein and data and instructions associated therewith are stored in respective storage devices, which are implemented as one or more non-transient machine or computer-readable or computer-usable storage media or mediums. Such computer-readable or computer usable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured single component or multiple components. The non-transient machine or computer usable media or mediums as defined herein excludes signals, but such media or mediums may be capable of receiving and processing information from signals and/or other transient mediums.

[0078] Example embodiments of the material discussed in this specification can be implemented in whole or in part through network, computer, or data based devices and/or services. These may include cloud, internet, intranet, mobile, desktop, processor, look-up table, microcontroller, consumer equipment, infrastructure, or other enabling devices and services. As may be used herein and in the claims, the following non-exclusive definitions are provided.

[0079] In one example, one or more instructions or steps discussed herein are automated. The terms automated or automatically (and like variations thereof) mean controlled operation of an apparatus, system, and/or process using computers and/or mechanical/electrical devices without the necessity of human intervention, observation, effort and/or decision.

[0080] It will be appreciated that any components said to be coupled may be coupled or connected either directly or indirectly. In the case of indirect coupling, additional components may be located between the two components that are said to be coupled.

[0081] In this specification, example embodiments have been presented in terms of a selected set of details. However, a person of ordinary skill in the art would understand that many other example embodiments may be practiced which include a different selected set of these details. It is intended that the following claims cover all possible example embodiments.

1. An apparatus comprising at least one processor and at least one memory including computer program code,

the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

based on at least one utterance of a pass-phrase and predetermined scoring information comprising pre-

determined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

provide for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

2. The apparatus of claim **1**, wherein the apparatus is configured to base said spoken pass-phrase suitability on at least two utterances of the pass-phrase and the pass-phrase suitability is also determined to be deficient based on a measure of spoken pass-phrase consistency comprising a difference between the at least two utterances being above a predetermined consistency threshold.

3. The apparatus of claim **1**, wherein the linguistic analysis comprises one or more of:

phonetic analysis, glottal voice source feature analysis, morpheme analysis, prosodic unit analysis, phonological analysis, syllable analysis, onset and rime analysis, articulatory features analysis and mora analysis.

4. The apparatus of claim **1**, wherein the linguistic elements comprise one or more of:

words, syllables, phonetic parts, prosodic units, patterns of two or more phonetic parts in the at least two utterances, patterns of two or more prosodic units in the at least two utterances.

5. The apparatus of claim **1**, wherein on determination that the pass-phrase suitability is deficient based on the pass-phrase score being below the predetermined pass-phrase score threshold, the apparatus is configured to provide for prompting of a user to change their pass-phrase.

6. The apparatus of claim **2**, wherein on determination that the pass-phrase suitability is deficient based on the difference between the at least two utterances being above a predetermined consistency threshold, the apparatus is configured to provide for prompting of a user to make one or more further utterances of the pass-phrase.

7. The apparatus of claim **2**, in which the apparatus is configured to generate a pass-phrase model from the at least two utterances of the pass-phrase, the pass-phrase model comprising a statistical description of the utterances of the pass-phrase, wherein the measure of spoken pass-phrase consistency comprises a difference between the model and a corresponding statistical description of at least one of the at least two utterances.

8. The apparatus of claim **7**, in which the measure of spoken pass-phrase consistency comprises a log-likelihood ratio.

9. The apparatus of claim **7**, wherein the pass-phrase model comprises one or more of: a hidden Markov based model, a Gaussian mixture model, i-Vector probabilistic linear discriminant analysis model and a neural network based model.

10. The apparatus of claim **1**, wherein the linguistic elements comprises phonemes and/or phones and the apparatus is configured to provide for segmentation of the one, or each, utterance of the pass-phrase into a plurality of individual phonemes and/or phones, and wherein each of the plurality of phonemes and/or phones is assigned its linguistic-element-score in accordance with the predetermined scoring information, the pass-phrase score based on the linguistic-element-scores for the plurality of phonemes and/or phones.

11. The apparatus of claim **10**, wherein and the pass-phrase suitability is also determined to be deficient based on an identification of insufficient linguistic elements that have a linguistic-element-score above a distinctiveness threshold using a minimum distinctiveness threshold.

12. The apparatus of claim **1** in which the apparatus is configured to provide for the spoken pass-phrase suitability determination as part of an enrolment procedure in which a user provides a spoken pass-phrase for future use to authenticate the identity of the user.

13. The apparatus of claim **1** in which the apparatus comprises at least part of: portable electronic device, a mobile phone, a Smartphone, a laptop computer, a desktop computer, a tablet computer, a personal digital assistant, a digital camera, a smartwatch, a non-portable electronic device, a monitor, a household appliance, a smart TV, a server, or a module/circuitry for one or more of the same.

14. A method comprising;

based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

providing for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

15. A computer readable medium comprising computer program code stored thereon, the computer readable medium and computer program code being configured to, when run on at least one processor having memory, perform at least the following:

based on at least one utterance of a pass-phrase and predetermined scoring information comprising predetermined linguistic-element-scores attributable to one or more linguistic elements that form at least part of each of the at least one utterance,

providing for spoken pass-phrase suitability determination wherein the at least one utterance is assigned a pass-phrase-score based on linguistic analysis in which one or more linguistic elements identified in said utterances are assigned their corresponding linguistic-element-score from the predetermined scoring information, the pass-phrase score based on the one or more linguistic-element scores of the, identified, linguistic elements, wherein the spoken pass-phrase suitability is

determined to be deficient at least based on the pass-phrase score being below a predetermined pass-phrase score threshold.

* * * * *

End-to-end automatic speaker verification with evolving recurrent neural networks

Giacomo Valenti [1,2]*, Adrien Daniel*[1]*, Nicholas Evans* [2]

[1] NXP Semiconductors, Mougins, France
[2] EURECOM, Biot, France

giacomo.valenti@eurecom.fr, adrien.daniel@gmail.com, evans@eurecom.fr

Abstract

The state-of-the-art in automatic speaker verification (ASV) is undergoing a shift from a reliance on hand-crafted features and sequentially optimized toolchains towards end-to-end approaches. Many of the latest algorithms still rely on frame-blocking and stacked, hand-crafted features and fixed model topologies such as layered, deep neural networks. This paper reports a fundamentally different exploratory approach which operates on raw audio and which evolves both the weights and the topology of a neural network solution. The paper reports what is, to the authors' best knowledge, the first investigation of evolving recurrent neural networks for truly end-to-end ASV. The algorithm avoids a reliance upon hand-crafted features and fixed topologies and also learns to discard unreliable output samples. Resulting networks are of low complexity and memory footprint. The approach is thus well suited to embedded systems. With computational complexity making experimentation with standard datasets impracticable, the paper reports modest proof-of-concept experiments designed to evaluate potential. Results equivalent to those obtained using a traditional GMM baseline system and suggest that the proposed end-to-end approach merits further investigation; avenues for future research are described and have potential to deliver significant improvements in performance.

1. Introduction

Deep learning approaches to automatic speaker verification (ASV) have emerged in recent years and are now at the state of the art. Deep learning techniques have been explored in the context of: feature extraction [1, 2]; the learning of posteriors in a joint factor analysis framework [3]; the extraction of phonetically-aware frame posteriors as a replacement for the universal background model in an i-vector framework [4]; an alternative to i-vectors within a probabilistic linear discriminant analysis (PLDA) framework [5]; the estimation of hidden Markov model state posterior probabilities [6, 7]; PLDA back-end scoring [8].

A common characteristic to the above works is the use of deep learning techniques as a means of replacing specific, and often single elements of a more complex toolchain. While it has demonstrated the benefit of deep learning, this work may not be capitalizing on the true potential whereby deep learning is applied in a so-called end-to-end approach; current techniques, *e.g.* [9], still rely on hand-crafted features or pre-determined topologies whereas evolutive learning techniques can facilitate the application of neural networks to raw inputs and automatically optimize the topology according to the task at hand.

A growing number of attempts have been made to overcome the reliance on hand-crafted features. Most operate on spectral representations, *e.g.* [10, 11, 12]. While spectral representations stem from a linear transformation of the raw audio, and thus serve as an equivalent representation, these solutions still rely upon the frame-blocking of speech signals into fixed-length windows. While recurrent neural networks, *e.g.*, long short-term memory (LSTM) architectures, can exploit speech dynamics [13], frame-blocking remains a perhaps-questionable constraint. In this sense, the avoidance of frame-blocking may offer some potential to improve on current approaches. While the literature shows successful attempts to apply deep learning techniques to raw audio, *e.g.* [14, 15, 16, 17, 18], these works relate to speech and emotion recognition in addition to spoofing detection. To the best of the authors' knowledge, there is no equivalent work in ASV.

Also characteristic to almost all attempts to use deep learning for ASV is the use of pre-determined topologies, namely topologies chosen manually and empirically optimized. Most deep learning solutions involve a layered, hierarchical approach [19, 20] in which the number of layers, their connectivity (local or full), the number of units per layer and their activation function (linear, rectified, *etc.*) are all predetermined. Research from beyond the field of speech processing, *e.g.* [21], suggests that the use of pre-determined topologies may be a limitation. Studies related to image classification, for example, show that topologies can be learned automatically [22], albeit still in hierarchical fashion.

Solutions to these limitations are already available. A particular class of techniques known as topology and weight evolving artificial neural networks (TWEANNs) [23] use genetic learning algorithms to optimize not only the weights of a network but also its topology. This paradigm embraces the principles of natural evolution and selection and allows connections between any units, thereby completely avoiding the notion of hierarchical layers.

Motivated by recent work on evolving recurrent neural networks for audio processing and classification [24], by the increasing popularity of end-to-end learning [9, 19, 25, 26] and to address the limitations of hand-crafted features and pre-determined topologies, this paper reports what is believed to be the first application of TWEANNs to ASV. The approach operates directly on unprocessed, raw audio which is treated subsequently by evolving network structures before final classification, making for a *truly* end-to-end pipeline.

Accordingly, the objective of the work presented in this paper was not to outperform the existing state of the art, but

This work was completed while A. Daniel was still at NXP Semiconductors.

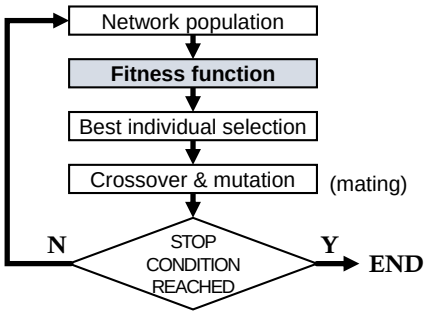


Figure 1: *An illustration of one iteration of evolution: the performance of each network in a population is assessed by the means of a fitness function and the* best *individuals are selected to form a new generation of networks.*

more specifically to investigate the longer term potential of the idea. The paper reports investigations with a particular form of TWEANN algorithms known as neuroevolution of augmenting topologies (NEAT) [27].

With the computational complexity of the algorithm far exceeding that of the established approaches to ASV (training only), experimentation with standard databases is currently impracticable. Implementation of the approach using efficient graphics processors is also far from being straightforward. In order to assess the potential of the idea, the paper reports modest proof-of-concept experiments designed to evaluate potential. The authors fully accept that the statistical significance afforded by such analysis is limited. With the algorithm representing something of a departure from current research directions and with results showing potential, the authors have elected to submit, admittedly early, the idea to the scrutiny of the scientific community.

Section 2 introduces the NEAT algorithm and its application to acoustic signals. Section 3 describes its adaptation and additional developments which are necessary such that the algorithm can be applied successfully to the ASV task. Experiments and results are described in Section 4. Conclusions are presented in Section 5.

2. Neuroevolution of augmenting topologies

The NEAT algorithm was introduced by Stanley and Miikkulainen in 2002 [27]. This section describes the main ideas behind the original work and then its previous application to audio classification problems.

2.1. Original algorithm

At a higher level, NEAT is a classical neuro-evolution algorithm which evolves a population of solutions (networks or individuals) according to an iterative process and a defined fitness function. Each iteration produces a new generation of solutions and the fitness function controls which among them serve as a basis to produce the next generation of solutions as shown in Fig 1.

At a lower level, however, NEAT is quite unique. One crucial aspect centers around the incremental evolution of structure. Even if the algorithm does not incorporate an explicit measure of complexity, networks tend to remain comparatively simple in structure compared to deep neural network solutions. Topologies are augmented iteratively in order to introduce diversity through the addition of new nodes and connections

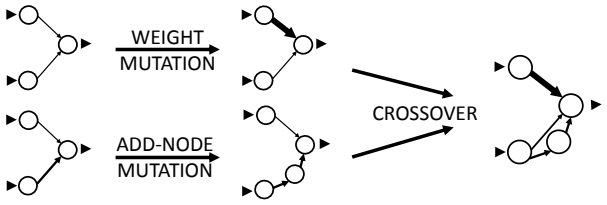


Figure 2: *Mutation of weight (here symbolized by connection thickness), node adding and crossover: the three forms of network evolution.*

thereby following a *complexifying* principle. This is achieved through the usual biological analogies of mutation and crossover. These processes, in addition to that of weight mutation, are illustrated in Fig. 2.

NEAT provides an elegant and efficient solution to a number of previously identified technical challenges such as the permutations problem [28]. These are addressed through the introduction of a genotype direct encoding scheme that features *historical markings* which track structural augmentations (see Fig. 3). Historical markings also serve a crucial purpose when performing crossover, as they provide a systematic means to align genes. Topology diversity is ensured by *speciation*, another biological analogy which protects structural innovation (*i.e.* by selecting the fittest networks within the same species).

With TWEANNs, weight and structural changes occur at random (within set boundaries) during evolution through mutation. The fitness function is an evaluation metric that aims to reward network changes which lead to improved performance. Hence the optimization process is not based on gradient descent and back-propagation, but instead consists in evaluating each network in the population according to the fitness function. To do so, training inputs are processed through the network exactly as they would be during inference, thereby yielding output values that serve to evaluate the network fitness. Then the best performing (i.e., the fittest) networks of the current population are selected to produce offspring for the next generation.

Since its conception, NEAT has been applied successfully to a multitude of tasks such as bipedal locomotion [29] and automated computer game playing [30]. NEAT continues to attract attention; shortly before the submission of this article, the authors became aware of recent, successful attempts to utilize NEAT for audio-related tasks such as audio effect creation [31] and sound event detection [32]. In view of the computational demands, however, in the former work NEAT was applied using a variety of classic spectral/cepstral frame-based features instead of raw audio, while the latter used wavelet representations.

2.2. Application to audio classification

Whereas the use of some form of frame-blocked spectral or filter-bank representation is characteristic to all previous work, Daniel [24] reported the first application of NEAT to audio classification which operates directly on time-domain inputs. NEAT is applied with networks constrained to a specific input/output setup and propagation scheme. As illustrated in Fig. 4, inputs consist of one or more streams of raw audio. Each stream is mapped to an input unit and is propagated through the network

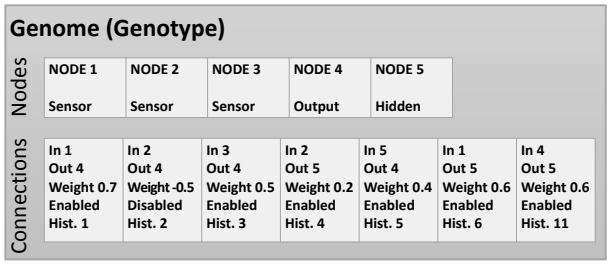


Figure 3: *A NEAT genotype is a direct and self-contained textual representation of a unique network, which contains (as in nature) more information than that which can be observed in the resulting structure. Figure reproduced with permission from [27].*

sample-by-sample with one activation step for every sample. An additional bias unit is set and held to unity. Network outputs consist of one or more *score* units whose outputs y_d are multiplied by the output of a binary *gate* unit y_r. Except for the score and gate output units, which have identity and binary step activations respectively, all units have rectified linear activation functions. The rate of the output (of any unit) is identical to that of the input, hence the networks perform one activation step per sample (see Fig. 4). In fact, the score output can be viewed as a new audio signal, the result of the network learning and applying to the input a transformation defined by the class to which the input belongs. The gate will thus evolve to discard *output* scores which are deemed to be unreliable, so that the network places emphasis on samples that are most helpful to discriminate between different audio classes. Alternatively, the gate can be replaced by a *reliability output* yielding a non-negative, non-binary weighting factor y_r. The operation of the gate/reliability output is similar in principle to that of attention mechanisms [33] which have been applied previously to speech recognition [34]. For each time sample i, the weighted mean over K samples of the product of y_d and y_r yields final weighted score y_w:

$$y_w[i] = \frac{\sum_{j=0}^{K-1} y_d[i-j] \times y_r[i-j]}{\sum_{j=0}^{K-1} y_r[i-j]} \quad (1)$$

The behavior of each network is assessed according to a generic squared-error-based fitness function F:

$$F(y_w, g) = 1/\left[1 + \sum_{i=0}^{N-1} \left(g[i] - y_w[i]\right)^2\right] \quad (2)$$

which reflects the distance between N weighted scores y_w and a ground truth signal g of classification labels, *e.g.* 0 or 1, making for a supervised approach. Connections can be made freely between *any* pair of units. As a result, evolved networks may contain cyclical unit connections (e.g. units connected to themselves or to other units which influence their input). This classifies NEAT structures as *recurrent* neural networks.

3. End-to-end automatic speaker verification

This section reports the application of NEAT to conceive a truly end-to-end ASV system. In the work of [24], all networks are constrained to share the common setup and propagation scheme illustrated in Fig. 4: there is one input stream, one bias, one output stream and a binary gate. The process described in Section 2.2 is applied to generate networks which distinguish between a given target speaker and a set of background speakers. Each iteration of the algorithm corresponds to one independent evolutionary process applied in speaker-dependent fashion. This process will produce a population of increasingly discriminative, speaker-dependent networks.

The evolutionary process is driven according to a new fitness function which is introduced below. Also described in this section is a mini-batch procedure which was found to be beneficial to the evolution process. Specific training and testing procedures are also presented.

3.1. Fitness function

The fitness function in Eq. 2 does not necessarily reward separation between class distributions, but rather proximity to ground truth scores (*e.g.* 0 and 1). This behaviour becomes an evident problem when, after several generations, the two classes have only a minimal degree of overlap: a distance-based fitness function would reward a network that pushes the bulk of the distributions farther apart, without necessarily correcting previous classification errors; conversely, a network which fully separates classes but which produces noisier distributions would be attributed less reward than another network which produces pure Gaussian, but slightly overlapped distributions. An early search for an alternative, better suited to classification tasks such as ASV, investigated a fitness function based on the equal error rate (EER). The EER, though, only reflects the reliability of a classifier at a single operating point, *i.e.*, a fixed threshold. The area under the receiver operating characteristic curve (AUROC), in contrast, gives a measure of reliability which is independent from the operating point; it reflects the probability that the network will give a randomly chosen target sample a higher score than a randomly chosen impostor sample [36]. With notably better results, all work reported in this paper was performed by replacing Eq. 2 with an AUROC function calculated using the *trapezoid rule* [37].

3.2. Mini-batching

Inspired by a similar approach used in the *stochastic gradient descent* algorithm [38] to avoid over-fitting and convergence to local-optima, training is performed with a mini-batch process. The mini-batch process ensures that each generation of networks is trained using a different subset of data. This strategy promotes novelty during evolution since the training objective is changed every iteration. The same strategy also encourages generalization, namely networks which perform well across inter-session data. Finally, mini-batching also helps to reduce computational demands.

Each mini-batch consists of a fraction M_t of total target data and a fraction M_i of total impostor data. By way of example, with M_t=M_i=100%, every training iteration is performed using the *same* data; there is no mini-batching. With M_t=M_i=50%, training data is randomly shuffled and partitioned into two mini-batches. They are used in two subsequent iterations after which this process is repeated.

3.3. Training

The initial generation contains a population of 150 minimal, perceptron-like networks, each of which is configured according to the setup described in Section 2.2. The network weights

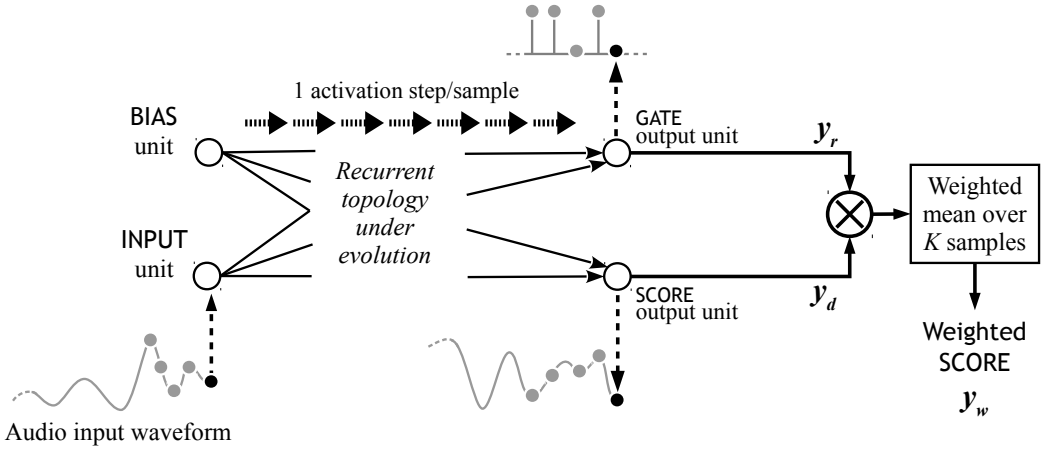


Figure 4: *End-to-end setup and propagation scheme for audio classification. There is one activation step per input sample; the output rate is the same as that of the input.*

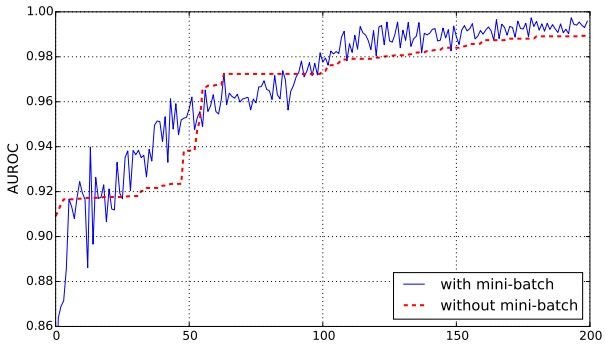


Figure 5: *An illustration of evolution measured in terms of fitness (AUROC) for the fittest network of each generation. The solid blue profile illustrates the AUROC with mini-batch training whereas the monotonic red profile shows the AUROC without mini-batch training.*

are randomly initialized and constrained within a $[-4, 4]$ range. Audio signals are normalized to within [-1, 1] in order to prevent saturation. This is more likely when using rectified linear activation functions as opposed to sigmoids, as in the original work. Rectified linear activation functions were found to be more efficient while giving similar performance. Every network in a given population is trained with the same mini-batch of data. Data containing either target or impostor speech is presented to each network in the form of non-contiguous segments of K samples. The system assigns to each segment a weighted mean score corresponding to $y_w[K-1]$ in Eq. 1. Networks are reset after the processing of each segment.

The fitness of each network is then determined according to the AUROC metric described in Section 3.1. The fittest networks of the population are then used to produce the next generation according to the procedure outlined in Section 2. The evolutionary process is applied iteratively until the fitness has converged.

Fig. 5 illustrates the evolution in fitness over 200 generations for an arbitrary target speaker. Each point on the graph corresponds to the population's fittest network for that generation. The solid blue profile illustrates evolution for the training procedure described above. Its non-monotonic nature is due to mini-batching; the data used at each iteration is different. The dashed red profile shows evolution with no mini-batching (M_t=M_i=100%); data used at each iteration is the same, hence the monotonic profile. While reducing processing time, mini-batching also results in faster learning.

3.4. Network selection for evaluation

Once training is complete, it is necessary to select and evaluate the single best network. First, the 10 best networks of each generation are identified according to the AUROC fitness function. Second, the performance of each of the 10 best networks from each generation is reassessed using the *full* training set. Since it gives a more intuitive interpretation of performance in a practical application, selection is performed using the application-neutral EER metric. The network which produces the lowest EER among the 10 is designated as the *generation champion*. Finally, the generation champion associated with the lowest EER is designated as the *grand champion*, and selected for evaluation. Evaluation is performed using an independent test set.

Aside from the fitness function and minor differences, this setup is also adopted in our own ongoing work in anti-spoofing [35].

4. Experiments

This section describes experiments which aim to test the potential of the end-to-end ASV system described in section 3.

4.1. Baseline system

The baseline system is a standard 64-component Gaussian mixture model (GMM) system [39]. Features are standard 19th order Mel-scaled frequency cepstral coefficients (MFCCs). These are appended with delta and double-delta parameters thereby giving features of 57 coefficients. Speaker models are derived from the maximum a posteriori adaptation of a universal background model (UBM). Scores are log-likelihood ratios given the speaker model and the UBM.

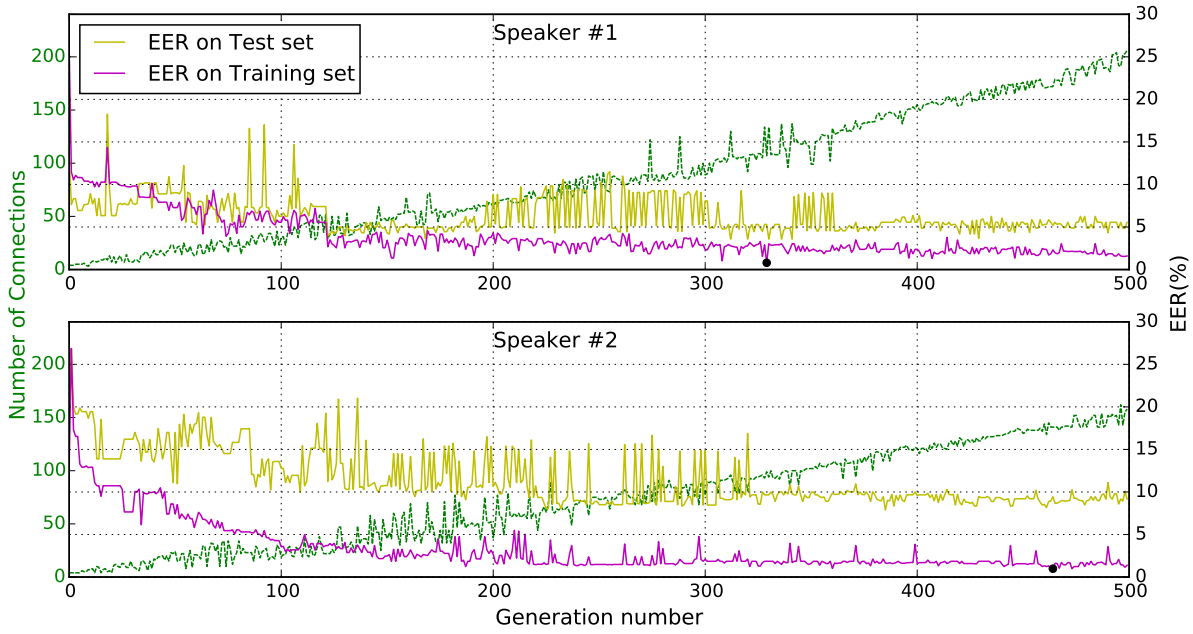


Figure 6: *Number of connections (green dashed profiles) and equal error rate (EER) of the first 500 generation champions for target speakers 1 (top) and 2 (bottom). EER profiles are shown for training data (magenta/dark) and testing data (yellow/light profiles). Black dots signify the grand champion, chosen according to the lowest-EER on the full training set.*

Table 1: *Results for the GMM and end-to-end systems in terms of EER for the training and test set for the two target speakers.*

	GMM Baseline		**End-to-end system**	
	Training	Test	Training	Test
Speaker #1	0%	9.52%	0.79%	5.30%
Speaker #2	0%	6.90%	0.98%	9.44%

4.2. NXP database and experimental protocols

Experimentation with standard NIST Speaker Recognition Evaluation (SRE) datasets [40], RSR [41] or RedDots [42] are currently impracticable on account of computational complexity. Being consistent with the objective to evaluate the potential of the algorithm, the paper reports a set of proof-of-concept experiments using a non-standard, proprietary database of speech signals collected from 10 male speakers. Text content consists of 10 of the 30 Harvard sentences which comprise the TIMIT database [43]. Each speaker provides approximately 5-6 minutes of speech which is recorded in 9 sessions over the course of one month. Recordings were collected in a quiet office with a laptop at a sampling rate of 16 kHz and 16-bit precision. Utterances were normalized by the active speech level estimated according to the ITU-T P.56 standard [44].

Among the 10 speakers, 2 are enrolled as targets. The training set consists of 6 of the 10 sentences uttered by the target speaker and the first 5 impostors. The test set consists of the other 4 sentences uttered by the target speaker and the remaining 3 impostors, thus achieving considerable phonetic separation between sets. Total target training data amounts to approximately 3.5 minutes of speech per speaker. Total impostor training data is in the order of 14 minutes duration.

For the end-to-end system, target data is partitioned into two mini-batches (M_i=50%). Since impostor data is more plentiful, it is partitioned into five mini-batches (M_i=20%) and used as background data for the baseline system. The average training recording is 3.25 seconds long. For the assessment and testing of both systems, one trial corresponds to one entire recording. Accordingly, K is set to $3.25 \times 16000 = 52000$ samples for training, and to each trial length at testing. Audio files used by the GMM system are preprocessed with silence removal. This step is not performed for the end-to-end system.

4.3. End-to-end system: augmentation and generalization

The training process took 17 and 13 hours for speaker 1 and 2, respectively, on an 8-core CPU running at 3.5 GHz. Several NEAT parameters influence the training time, e.g. without mini-batching, and with an otherwise identical setup, training takes several days.

Results obtained according to the evaluation procedure described in Section 3.4 are depicted in Fig. 6. Results are illustrated independently for the two target speakers and for 500 generations. The solid magenta (dark) profile in each plot shows the EER obtained by each generation champion assessed using the training data. EER profiles exhibit the expected evolution trend, namely a steady decrease from above 30% to less than 5% within 150 generations. The lowest EERs obtained by grand champion networks are 0.79% for speaker 1 (generation 329) and 0.98% for speaker 2 (generation 464) marked by black dots. Solid yellow (light) profiles show EERs for generation champions assessed on test data. As expected, performance

on independent data is worse. Nonetheless, the selected grand champions are among the best performing networks on test data.

A summary of performance for both GMM and end-to-end systems is presented in Table 1. For the latter, results for both train and test datasets concern the grand champion network selected for each speaker. For the test set, grand champions yield EERs of 5.30% and 9.44%, whereas the GMM system delivers EERs of 9.52% and 6.90%. The gates of the grand champion networks prune an average of 46% of output data (in both speech and non-speech intervals) — the average for speaker 1 is 40% whereas that for speaker 2 is 53%. This percentage is consistently higher than than for the GMM system for which silence removal prunes an average of 35% of data. The effective behaviour of the gate was observed on a just few trials, depicting a periodic opening and closing as opposed to an energy- or amplitude-related activation. These findings show that (i) the performance of the end-to-end system is competitive with that of the GMM system and (ii) the two systems exploit data in a different way.

The upper green dashed profiles in Fig. 6 show the number of connections of each generation champion. As evolution proceeds, networks are steadily augmented with new nodes and connections. In general, network augmentations cause decreases in EERs for the training set, with 112 and 138 connections for speaker 1 and 2 grand champions, respectively. These networks are orders of magnitude less complex than usual, deep layered structures (*c.f.* $\sim 200k$ connections for the most compact model reported in [20]). Networks with such a reduced parameter space are inherently less prone to over-fitting since they do not have the capacity to learn a direct input-output correspondence.

5. Conclusions and future work

This paper reports an end-to-end approach to automatic speaker verification (ASV) based on the neuroevolution of augmenting topologies (NEAT) algorithm. In contrast to the existing state of the art, the proposed algorithm avoids the use of handcrafted features by processing raw audio and optimizes network weights and topologies in an entirely end-to-end fashion. Less complex topologies with a low memory footprint are well suited to embedded implementations. While reporting results for two speakers is not sufficient—and was neither intended—to provide a statistically reliable comparison between two systems, the proposed end-to-end approach is found to be at least competitive with a GMM baseline system.

These findings suggest that the end-to-end approach merits further attention and experimentation with larger, standard datasets. This work will require the reduction of computational efficiency; computational demands of the current algorithm make larger-scale experimentation impracticable.

In addition to the investigation of efficient implementations which exploit hardware acceleration, future work should consider non-binary gates for soft, rather than hard weighting of output score samples, and experimentation with longer duration training and testing. This work may well bring improvements in end-to-end system performance and/or expose application settings for which the proposed approach may excel.

6. References

[1] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in Interspeech, 2011, vol. 237, p. 240.

[2] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification, in INTERSPEECH, 2017, pp. 1542-1546.

[3] S. Dey, S. Madikeri, M. Ferras, and P. Motlicek, "Deep neural network based posteriors for text-dependent speaker verification," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5050-5054.

[4] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 1695-1699.

[5] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, Deep neural network embeddings for text-independent speaker verification, in INTERSPEECH, 2017, pp. 999-1003.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, 2012, pp. 30-42.

[7] Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," Acoustics, Speech and Signal Processing (ICASSP) IEEE International Conference on, 2014, pp. 185-189.

[8] H.-S. Lee, Y.-D. Lu, C.-C. Hsu, Y. Tsao, H.-M. Wang., and S.-K. Jeng, "Discriminative autoencorders for speaker verification", in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, 2017.

[9] A. Miguel, J. Llombart, A. Ortega, and E. Lleida, "Tied hidden factors in neural networks for end-to-end speaker recognition, in INTERSPEECH, 2017, pp. 2819-2823.

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

[11] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in Advances in neural information processing systems, 2009, pp. 1096-1104.

[12] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-End Attention based Text-Dependent Speaker Verification," preprint arXiv:1701.00562, Jan. 2017.

[13] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," preprint arXiv:1402.1128, 2014.

[14] D. Palaz, R. Collobert, and M. M. Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," eprint arXiv:1304.1018, 2013.

[15] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in INTERSPEECH, 2015, pp. 1-5.

[16] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in INTERSPEECH, 2014, pp. 890-894.

[17] G. Trigeorgis et al., "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, 2016, pp. 5200-5204.

[18] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection," in IEEE IAPR International Joint Conference on Biometrics (IJCB), 2017.

[19] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, 2016, pp. 5115-5119

[20] , "Locally-connected and convolutional neural networks for small footprint speaker recognition.," in INTERSPEECH, 2015, pp. 1136-1140.

[21] Zhang, B.-T. and Muhlenbein, H., "Evolving optimal neural networks using genetic algorithms with Occams razor" in Complex Systems, no. 7, 1993, pp 199-220.

[22] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning", in International Conference on Learning Representations (ICLR), 2017.

[23] D. Dasgupta and D. McGregor, "Designing application-specific neural networks using the structured genetic algorithm", In Proceedings of the International Conference on Combinations of Genetic Algorithms and Neural Networks, IEEE Press, Piscataway, New Jersey, 1992, pp 87-96.

[24] A. Daniel, "Evolving recurrent neural networks that process and classify raw audio in a streaming fashion", in INTERSPEECH, 2017.

[25] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1764-1772.

[26] Y. LeCun, Y. Bengio and G. Hinton,"Deep learning", in Nature, no. 521, 2015, pp 436-444.

[27] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," Evolutionary computation, vol. 10, no. 2, 2002, pp. 99-127.

[28] N. J. Radcliffe, "Genetic set recombination and its application to neural network topology optimisation," Neural Computing and Applications, no. 1, 1993, pp. 67-90.

[29] B. Allen and Petros Faloutsos. "Complex networks of simple neurons for bipedal locomotion", In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009.

[30] M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone, "A neuroevolution approach to general Atari game playing," IEEE Transactions on Computational Intelligence and AI in Games, vol. 6, no. 4, 2014, pp. 355-366.

[31] I. Jordal,"Evolving artificial neural networks for cross-adaptive audio effects," Masters Thesis, NTNU, 2017.

[32] C. Kroos and M. Plumbley,"Neuroevolution for sound event detection in real life audio: A pilot study," Detection and Classification of Acoustic Scenes and Events (DCASE) Proceedings, 2017.

[33] C. Olah, and S. Carter, "Attention and augmented recurrent neural Networks", in Distill, 2016.

[34] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," eprint arXiv:1508.01211v2, 2015.

[35] G. Valenti, H. Delgado, M. Todisco, N. Evans and L. Pilati, "An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks", Speaker Odyssey 2018.

[36] T. Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters, 27, 2006, pp. 861-874.

[37] J. A. C. Weideman, "Numerical integration of periodic functions: a few examples", The American Mathematical Monthly, no. 109, 2002, pp. 21-36.

[38] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT2010, Springer, 2010, pp. 177-186.

[39] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," Digital Signal Processing, vol. 10, no. 13, Jan. 2000, pp. 19-41.

[40] S. O. Sadjadi et al., "The 2016 NIST Speaker Recognition Evaluation," in INTERSPEECH, 2017, pp. 1353-1357.

[41] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases.," in INTERSPEECH, 2012.

[42] K. A. Lee et al., "The RedDots Data Collection for Speaker Recognition," in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[43] Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., Dahlgren, N., Zue, V., "Timit acoustic-phonetic continuous speech corpus linguistic data consortium. Philadelphia, PA, 1993.

[44] International Telecommunication Union, "ITU-T P.56", Series P: Terminals and Subjective and Objective Assessment Methods - Objective measurement of active speech level, 2011.

An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks

Giacomo Valenti[1,2], *Héctor Delgado*[2], *Massimiliano Todisco*[2], *Nicholas Evans*[2] *and Laurent Pilati*[1]

[1]NXP Semiconductors, Mougins, France ; [2]EURECOM, Biot, France

valenti@eurecom.fr, evans@eurecom.fr, delgado@eurecom.fr

todisco@eurecom.fr, laurent.pilati@nxp.com

Abstract

Research in anti-spoofing for automatic speaker verification has advanced considerably in the last three years. Anti-spoofing is a particularly difficult pattern classification problem since the characteristics of spoofed speech vary considerably and can never be predicted with any certainty in the wild. The design of features suited to the detection of unpredictable spoofing attacks is thus a staple of current research. End-to-end approaches to spoofing detection with exploit automatic feature learning have shown success and offer obvious appeal. This paper presents our efforts to develop such a system using recurrent neural networks and a particular algorithm known as neuroevolution of augmenting topologies (NEAT). Contributions include a new fitness function for network learning that not only results in better generalisation than the baseline system, but which also improves on raw performance by 22% relative when assessed using the ASVspoof 2017 database of bona fide speech and replay spoofing attacks. Results also show that mini-batch training helps to improve generalisation, a technique which could also be of benefit to other solutions to the spoofing detection problem.

1. Introduction

Automatic speaker verification (ASV) [1, 2] offers a convenient, reliable and cost-effective approach to person authentication. Voice-based authentication is nowadays used in a plethora of logical and physical access scenarios, e.g. for telephone banking or for smartphone logon [3]. Despite the success, vulnerabilities to spoofing (also known as presentation attacks) give reason for caution. Without adequate countermeasures, fraudsters can manipulate the normal operation of an authentication system by masquerading as genuine users and hence gain unauthorised access to protected resources or services. Vulnerabilities to presentation attacks are clearly inadmissible; in addition to the immediate security concerns, they undermine confidence in ASV technology.

It is known that ASV systems can be vulnerable to spoofing attacks in the form of impersonation, synthetic speech, converted voice and replay [4]. Impersonation (the imitation of a target speaker by another person) requires a certain skill and is generally considered to pose only a modest risk [5]. While the threats posed by synthetic speech and converted voice are potentially severe, given that their implementation requires specialist expertise, the actual risk may be relatively low. Replay attacks arguably present the greatest threat. Replay attacks involve the (surreptitious) capture and subsequent playback to the ASV system of a speech sample captured from a genuine speaker/user. The threat and risk posed by replay attacks is significant: replay attacks can be mounted easily with widely available, consumer-grade audio recording and playback devices (e.g. smart phones) and can be especially difficult to distinguish from genuine, bona fide speech samples.

Efforts to develop spoofing countermeasures, also known as presentation attack detection (PAD) systems, are now well under way; the study of spoofing countermeasures for ASV is today an established area of research [6]. The first competitive evaluation, namely the ASV spoofing and countermeasures (ASVspoof) challenge [7], was held in 2015. It promoted the development of countermeasures to protect ASV from voice conversion and speech synthesis attacks. The second edition of ASVspoof held in 2017 switched focus to the mitigation of replay attacks [8, 9, 10], the focus in this paper.

The many submissions to the ASVsoof 2017 challenge can be classified into one of two different approaches. The first set of approaches involves the combination of hand-crafted features with generative classifiers such as Gaussian mixture models (GMM) and i-vectors/PLDA systems, e.g. [11, 12, 13, 14, 15, 16, 17, 18]. The second of approaches explored the use of discriminative classifiers such as support vector machines (SVMs) and deep neural networks (DNNs) [13, 16, 19, 17, 20, 21, 18].

Deep learning techniques, in particular, proved to be especially successful, with five of the top ten performing systems submitted to the ASVspoof 2017 challenge employing some form of automatic feature learning and/or classification[1]. The work in [16] used convolutional neural networks for the automatic learning of features from magnitude spectrograms with a combination of convolutional and recurrent layers in an end-to-end solution.

End-to-end approaches to anti-spoofing have obvious appeal. In more traditional fields of speech processing, such as ASV, there is a substantial body of knowledge that has been acquired over decades of research. This knowledge has been exploited to design extremely effective hand-crafted features. Even in the case of ASV, though, automatic approaches to feature learning are bringing advances in performance [22]. Research in anti-spoofing is comparatively embryonic. The history of benchmarking evaluations spans only three years and the quest for better-performing, hand-crafted features is a staple of current research [1]. That the natural variation in spoofing attacks is so great makes hand-crafted feature design especially difficult. In the absence of an extensive body of background knowledge or proven features, and with the availability of large datasets of spoofed speech, automatic feature learning and end-to-end solutions present an opportunity to fast track progress.

[1]A summary of top submissions is available at http://www.asvspoof.org/slides_ASVspoof2017_Interspeech.pdf

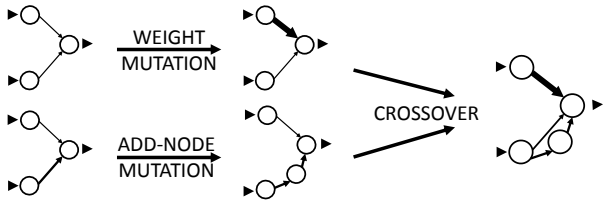


Figure 1: *Mutation of weight (here symbolized by connection thickness), node adding and crossover: the three forms of network evolution. Figure reproduced from [24].*

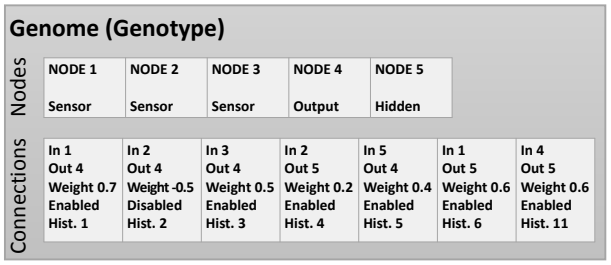


Figure 2: *A NEAT genotype is a direct and self-contained textual representation of an unique network, which contains (as in nature) more information than that which can be observed in the resulting structure. Figure reproduced with permission from [25].*

Motivated by the obvious appeal, by recent work in a similar direction [23] and by the success of the same technique for automatic speaker recognition [24], we have explored a particular approach to automatic feature learning in a truly end-to-end solution to anti-spoofing. It is based upon a class of algorithms known as topology and weight evolving neural networks (TWEANNs), specifically the neuroevolution of augmenting topologies (NEAT) algorithm proposed in [25]. The novel contributions in this paper are four-fold: (i) we present the first application of neuroevolutive learning to the anti-spoofing problem; (ii) we propose a fitness function that is better adapted to audio classification problems; (iii) we demonstrate the merit of automatic learning and end-to-end optimisation in tackling the so-called *generalisation* problem, namely solutions that do not generalise well to test data containing spoofing attacks different to those encountered in training and development, and (iv) we demonstrate that the proposed approach not only improves on generalisation, but that it also brings a significant improvement to the ASVspoof 2017 baseline results.

The remainder of the paper is organised as follows. Section 2 provides an overview of the NEAT algorithm and describes recent work that facilitates its application to audio classification tasks. Section 3 introduces a new fitness function tailored to the anti-spoofing problem. Experimental setup and results are the focus of Sections 4 and 5. Conclusions and ideas for future research are presented in Section 6.

2. NEAT

This section introduces the neuroevolution of augmenting topologies (NEAT) algorithm and describes its application to acoustic signals and their classification. Also described is a modification to the fitness function which was found to give better performance when NEAT was applied to audio classification tasks. The focus of this section is on *past* work. New contributions are the focus of Section 3.

2.1. Original work

The NEAT algorithm was introduced by Stanley and Miikkulainen in 2002 [25]. In similar fashion to other topology and weight evolving neural network (TWEANN) approaches, NEAT is a particularly elegant algorithm which exploits the appeal of both genetic algorithms and neural networks. The NEAT algorithm is initialised with a pool of candidate networks, all potential solutions to a given classification task. Inputs may be data samples or features whereas outputs are some form of score. The pool of networks evolves iteratively over many iterations, with the pool of networks within one iteration forming a generation of solutions. At each iteration, networks can mutate through the addition of new nodes or connections, the modification of connection weights or upon a crossing over of a pair of different networks (see Fig. 1).

In order to avoid confusion, it is stressed that TWEANNs make no use of backpropagation or gradient descent algorithms during training. Networks evolve only as a result of mutation according to the processes illustrated in Fig. 1. A measure of performance is required to control network selection and evolution. Performance is gauged according to the *fitness function*.

Since NEAT operates on a set of minimal initial structures and *augments* complexity gradually at each generation in order to solve the problem in hand, even fully evolved networks tend to be considerably less complex than typical deep neural network solutions. The relatively simple structure of NEAT networks means that they are well suited to embedded applications.

Network characteristics are described in the form of a genotype with *direct encoding*. The genotype is a compact representation of the structure (units and connections) and associated parameters (weights). The information encoded within identifies a unique individual (see Fig. 2). The chronological sequence of structural changes that occur between generations are recorded in the form of *historical markings*. In resolving the so-called structure permutation problem [26] they provide an elegant means of representing gene alignment which dictates which networks among a population are compatible for crossover.

Evolution is controlled according to the concept of so-called *fitness*. The fitness function is used to determine which networks within a current generation will contribute to form the next generation of networks (see Fig. 3). Fitness is evaluated according to the similarity between classification results with the labels from suitable quantities of training data. The NEAT algorithm is hence a form of supervised learning.

Structural innovation is fuelled by protecting a proportion of networks within a population that may not have the highest fitness. These networks may nonetheless have potential to produce better performing networks in later generations. This is achieved by clustering the population of networks into sets of *species* according to a measure of compatibility encoded in the genotype. At every iteration, all networks within the new population are assigned to the species with which they are most compatible. In the event that one or more networks are incompatible with the current set of species, then a new species is created. The best individuals belonging to each species are then

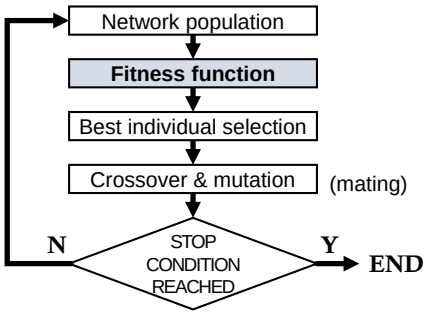


Figure 3: *An illustration of one iteration of evolution: the performance of each network in a population is assessed by the means of a fitness function and the* best *individuals are selected to form a new generation of networks. Figure reproduced from [24].*

selected, meaning networks compete for the best fitness within a niche. This concept of *speciation* serves to protect novelty and diversity within the population which hence has greater potential as a whole to solve the problem in hand.

NEAT has been applied successfully to a multitude of tasks such as bipedal locomotion [27], automated computer game playing [28], as an approach to general acoustic classification [29], audio effect generation [30] and sound event detection [31]. The work in [29] was the first to apply NEAT to the classification of raw audio.

2.2. Application to raw audio

A high-level overview of the framework proposed in [29] by which NEAT can be applied to the processing and classification of raw audio signals is illustrated in Fig. 4. Inputs consist of a bias unit (set to unity) and an input unit which is fed sample-by-sample by a raw audio signal. The latter is propagated through the network at one activation step per sample. There are two output units, a *gate* unit y_r and a *score* unit y_d, each of which produce one output sample per input sample. The network topology between inputs and outputs is of the form illustrated in Fig. 1. It is naturally recurrent in nature; there is no notion of hierarchical layers and no restrictions on links between units. With the exception of score and gate output units which have identity and binary step activation functions respectively, all units have rectified linear activation functions.

Connections can be made freely between *any* pair of units. As a result, evolved networks may contain cyclical unit connections (e.g. units connected to themselves or to other units which influence their input). This classifies NEAT structures as recurrent neural networks.

The product of the gate and score output units is averaged over K samples[2], thereby producing a weighted score y_w:

$$y_w[i] = \frac{\sum_{j=0}^{K-1} y_d[i-j] \times y_r[i-j]}{\sum_{j=0}^{K-1} y_r[i-j]} \quad (1)$$

where i is the sample index and where the gate output y_r is the weight. As proposed in [29], the weighting produced by the gate can be continuous, or may be constrained to a binary weighting $\{0,1\}$. While the behaviour of the gate is learned automatically, it will act naturally as a form of attention mechanism [32, 33], i.e. to emphasise the most salient output scores.

[2]The work in [29] proposed a flexible streaming/segmental or file-based approach were the value of K is adjusted accordingly. All work reported in this paper relates to a file-based processing approach where K is set to the number of samples in a file.

2.2.1. Fitness estimation

As in the original work, network performance is measured through a fitness function. The fitness function is key since it is used as a factor in the control of the evolutionary process. The work in [29] used a generic squared-error-based fitness function defined according to:

$$F = 1/\left[1 + \sum_{i=0}^{N-1} (g[i] - y_w[i])^2\right] \quad (2)$$

where g is a ground truth signal of classification labels, *e.g.* 0 and 1. The summation over N reflects the difference between labels and averaged scores, i.e. the inverted error gives a measure of reliability, or fitness.

Our own investigations using the NEAT algorithm for an automatic speaker recognition task [24] showed that the fitness function in Eq. 2 is not sufficiently informative as a means of guiding evolution. Eq. 2 reflects the average proximity of network scores to ground truth labels, rather than *classification* reliability. The latter is often measured with the application-neutral equal error rate (EER). Use of the EER was also found to be sub-optimal; it reflects the reliability of a classifier at a single operating point, *i.e.*, a fixed threshold.

Being independent to a specific operating point, the receiver operating characteristic (ROC) profile is a more informative measure of classifier reliability. ROC profiles may be reduced to a single scalar by measuring the so-called area under the ROC (AUROC) [34]. The AUROC is well tailored to classification as it is proportional to the ability of a classifier to attribute higher scores to positive trials than to negative trials. The work in [24] reports the replacement of Eq. 2 with an AUROC function calculated using the *trapezoid rule* [35].

2.2.2. Mini-batching

Mini-batch training can be used [24] to manage computational effort and to avoid over-fitting. Mini-batching is employed in a similar manner as with the *stochastic gradient descent* algorithm [36] whereby each iteration of training is performed with different batches of data, each a subset of the training partition. The learning of each generation with a different subset of training data promotes network novelty, reduces computation and encourages the evolution of networks that will have better potential to generalise well to unseen data.

The work in [24] defines a pair of mini-batch parameters M_t and M_i. They represent the fraction of available target and impostor data used for each step for the mini-batch training of an automatic speaker verification system. As an example, the setting of both parameters M_t=M_i=100% is equivalent to no mini-batching, with every generation being fed with the full partition of training data. In contrast, the setting of M_t=M_i=50% implies two mini-batches each comprising half of the available target and impostor training data. In this case, the composition of the mini-batches is reset and re-partitioned at random for *every other* generation.

Given the focus of this paper upon anti-spoofing rather than automatic speaker verification, notations M_t and M_i are replaced with M_b (bona fide speech) and M_s (spoofed speech). This notation is adopted throughout the remainder of this paper.

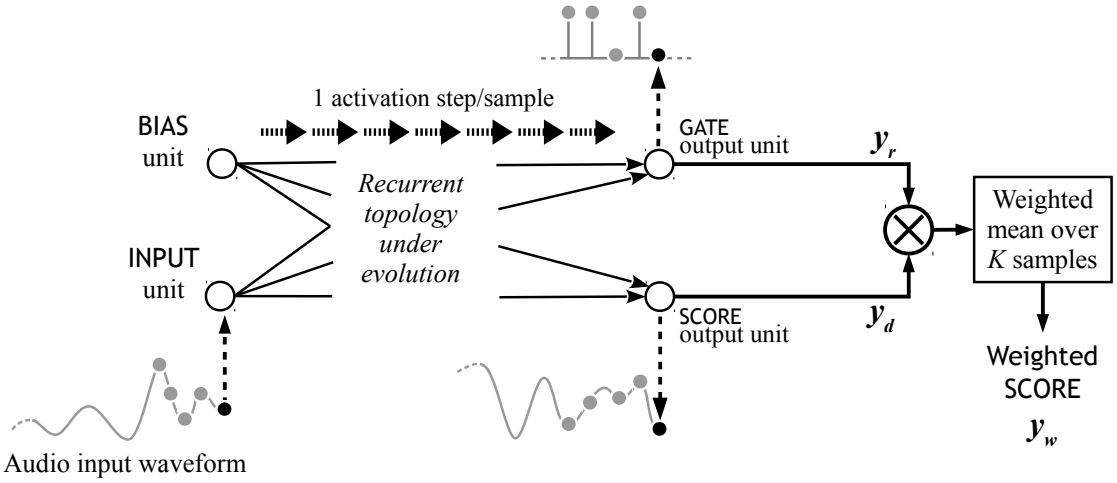


Figure 4: *End-to-end setup and propagation scheme for audio classification. Figure reproduced from [24].*

3. End-to-end anti-spoofing

This section describes the adaptation of the NEAT approach to the anti-spoofing problem. It encompasses the novel contributions claimed in this paper. These comprise a new fitness function which was found to give improved performance in the case of anti-spoofing, in addition to training and optimisation (network selection) procedures.

3.1. Ease of classification[3]

Experiments to evaluate the performance of NEAT for anti-spoofing using the previously reported fitness functions [29, 24] showed a tendency to oscillate around local optima, namely networks in subsequent generations that correct previous classification errors while introducing new ones. Such oscillations can be avoided by using an enhanced fitness function which rewards *progress* rather than raw performance. Progress infers better networks which correct previous classification errors *without* introducing new ones.

An expression for fitness which rewards progress requires the definition of a measure of segment (file) classification ease. Intuitively, this is proportional to how high or how low is the score for segment s compared to the average impostor (spoofed) or target (bona fide) scores respectively; For every network n and **bona fide** segment s with score θ_s, the classification ease is given by:

$$l_{s,n} \leftarrow 1 - \frac{\#\{\textbf{spoofed} \text{ segments with score} > \theta_s\}}{\#\{\textbf{spoofed} \text{ segments}\}} \quad (3)$$

where the right-most term is akin to the false acceptance rate for the given threshold. Conversely, for every **spoofed** segment with score θ_s, the classification ease is given by is given by:

$$l_{s,n} \leftarrow 1 - \frac{\#\{\textbf{bona fide} \text{ segments with score} < \theta_s\}}{\#\{\textbf{bona fide} \text{ segments}\}} \quad (4)$$

where the right-most term is now akin to the false rejection rate for the given threshold. A *pooled* measure of the classification ease may then be obtained by averaging the classification ease over the number G of networks in the population:

$$p_s \leftarrow \frac{\sum_n l_{s,n}}{G} \quad (5)$$

where $l_{s,n}$ is set according to Eqs. 3 or 4 depending on whether segment s is a bona fide or spoofed respectively. A measure of network fitness F is then estimated across all segments accordingly to:

$$F = \frac{\sum_s l_{s,n}(1 - p_s)}{\sum_s (1 - p_s)} \quad (6)$$

where $(1 - p_s)$ acts to weight the contribution of the classification ease for segment s, and network n. This approach to fitness estimation is from here on in referred to as the ease of classification (EOC).

According to Eq. 6, the correct classification of segments that were already correctly classified by networks in an earlier generation thus contributes little to the estimation of fitness for networks in the subsequent generation; there is little reward for learning what was already known by an earlier generation. The EOC approach to fitness estimation steers evolution to classify correctly a diminishing number of *difficult* segments.

3.2. Training

The size of each population is fixed across generations and set to 150 networks. The algorithm is initialised (generation zero) with 150 minimal perceptron-like networks, all of which share the common setup described in Section 2.2. All input signals are normalised to within a range of [-1,1]. The choice of rectified linear unit activation functions results in faster processing, but also increases the chances of saturation. The random initialisation of weights within a $[-4, 4]$ range was found to manage the risk of saturation.

Experiments were conducted with both AUROC (Section 2.2) and EOC (Section 3.1) fitness functions, with and without mini-batching. Audio signals containing either bona fide or spoofed speech are fed to each network segment-by-segment (file-by-file) and the network is trained in order to distinguish between the two. The AUROC fitness function is evaluated with

[3]The EOC fitness function was developed in collaboration with Adrien Daniel while he was employed at NXP Semiconductors.

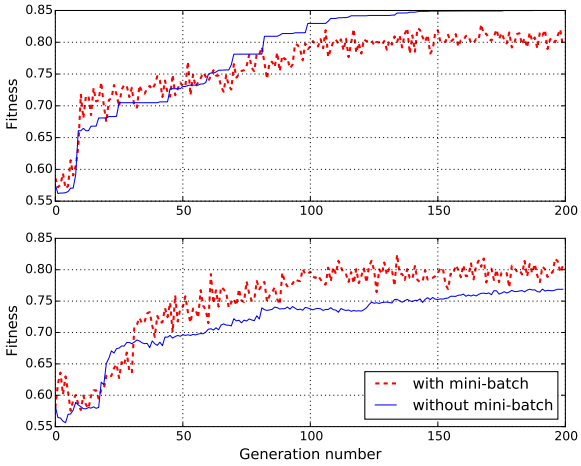


Figure 5: *An illustration of fitness evolution for the fittest network of each generation when using AUROC (top) and EOC (bottom) fitness functions. The dashed red profiles illustrates the fitness evolution with mini-batch training whereas the mostly-monotonic blue profiles shows the fitness without mini-batch training.*

K in Eq. 1 set to the number of samples in each file. All networks are reset after the processing of each file.

At each iteration (generation), a subset of the fittest (best performing) networks among each species is determined and used to evolve the next generation of networks according to the procedure outlined in Section 2.1. Evolution proceeds either until the fitness converges or until a pre-determined maximum number of generations is reached.

Fig. 5 depicts the improvement in network fitness (vertical axis) over 200 generations (horizontal axis). Illustrated is the evolution in fitness for four different configurations: the two fitness metrics AUROC and EOC, both with and without mini-batch training ($M_b = 25\%$ and $M_s = 33\%$). Each point on each profile shows the fitness of the single, fittest network among the 150 in the population. In both graphs the dashed red curves relate to mini-batch training. Neither profile is monotonic since the data changes at each generation. Conversely, solid blue curves show fitness without mini-batch training ($M_b = M_s = 100\%$), hence the largely monotonic profiles (the fitness of EOC optimised networks is not strictly monotonic on account of the different weights applied to each segment during fitness estimation, as described in Section 3.1).

Profiles in Fig. 5 show that mini-batching is of more benefit when used with the EOC fitness function. Changes in training data can be interpreted as optimisation towards a moving target. This fuels novelty instead of over-fitting to a fixed training set. These observations would suggest a potential for better generalised spoofing detection. It should be noted, however, that the final objective is not higher fitness for training data, but the *classification reliability* assessed using test data.

3.3. Testing

Networks with high measures of fitness may not necessarily be those which give the best performance in terms of the spoofing detection EER. This is especially true when using mini-batch since one random subset of training data could be fortuitously easier than another subset (or indeed the full set). In addition, measures of fitness derived using the EOC fitness function may not be especially well correlated with classification performance; increases in EOC reflect the learning of new information rather than raw performance.

As a result, the fittest network identified from training may not be that which gives the lower EER. In order to observe the evolution in classification performance, the 10 best networks of each generation identified using the fitness function are evaluated using development data and with an EER metric.

The single network with the lowest EER within each group of 10 is named the *generation champion* and the overall lowest EER network among the set of generation champions is denoted the *grand champion*. The latter is selected for testing/evaluation where it is used without further modification.

4. Experimental setup

This section describes the database, protocol and metric used for all experiments reported in this paper. Also described is the baseline system and specific configuration details for the proposed end-to-end approach to anti-spoofing.

4.1. Database, protocol and metric

Experiments were performed using Version 2.0[4] of the ASVspoof 2017 database [37]. The database originates from the RedDots database[5] which was collected by volunteers from across the globe using mobile devices, in the form of smartphones and tablet computers. While the RedDots database was collected to support research in text-dependent automatic speaker verification, the ASVspoof 2017 database was adapted from it in order to support research in anti-spoofing. It contains sets of bona fide (genuine) and replayed speech [38, 39, 9]. In order to simulate replay spoofing attacks, the bona fide partition of the ASVspoof 2017 database was replayed and then recaptured using a variety of different loudspeakers and recording devices in heterogeneous acoustic environments.

The standard protocol relates to a partition of the database into training, development and evaluation subsets, details of which are presented in Table 1. The three subsets are mutually disjoint in terms of speakers and of data collection sites. Experiments reported in this paper were performed with the extended protocol whereby both training and development were performed with pooled training and development partitions (train+dev). The evaluation subset contains data collected using 57 replay configurations, 49 of which differ to those used in the collection of the training and development subsets. Differences in replay detection performance between the training/development and evaluation subsets serve to gauge the generalisation of spoofing countermeasure solutions.

The ASVspoof 2017 evaluations assessed the performance of spoofing countermeasures in isolation to automatic speaker verification. The standard metric is the application-independent equal error rate (EER). It is used for all assessments reported in this paper.

4.2. Baseline

The ASVspoof 2017 Version 2.0 database was released in order to correct data anomalies detected subsequent to the official evaluation. Being released in 2018, the only published re-

[4] http://dx.doi.org/10.7488/ds/2301

[5] https://sites.google.com/site/thereddotsproject/

Table 1: *Statistics of the ASVspoof 2017 database version 2.*

Subset	# spk	# replay sessions	# replay configs	# utterances bona fide	# utterances replay
Training	10	6	3	1507	1507
Devel.	8	10	10	760	950
Eval.	24	161	57	1298	12008
Total	42	177	61	3566	14466

sults relating to Version 2.0 are those for the official ASVspoof 2017 baseline system[6]. It uses a constant Q cepstral coefficient (CQCC) [40, 41] frontend and a traditional Gaussian mixture model (GMM) back-end [42, 43]. Classifier scores are computed as the log-likelihood ratio for the test utterance given bona fide and replayed speech models. This paper considers only the extended protocol baseline for which training and development are performed using pooled training and development dataset (train+dev). Baseline results for the extended protocol are presented to the top of Table 2.

4.3. End-to-end anti-spoofing

The end-to-end algorithm described in this paper was applied to distinguish between bona fide and spoofed speech. All networks are configured according to the common setup described in Section 2.2 and as depicted in Fig. 4. Experiments were conducted with four different configurations comprising AUROC (Section 2.2.1) and EOC (Section 3.1) fitness functions with and without mini-batch training (Section 2.2.2). Configurations in which mini-batch is adopted are labeled with m (see Table 2). Each configuration was run for 500 generations.

When applied, mini-batch training is performed with bona fide speech partitioned into four mini-batches (M_b=25%) of approximately 17 minutes each. Spoofed data is partitioned into three mini-batches (M_s=33%), approximately 21 minutes each. The discrepancy between bona fide and spoofed speech is due to the greater variation in spoofed speech, the reliable modelling of which requires greater quantities of data in each batch.

Once the training of a generation is completed, the performance of networks for that generation is assessed according to the procedure described in Section 3.3. This assessment is performance using pooled training and development partition data (see Section 4.1).

5. Experimental results

This section describes experimental results, starting with an illustration of the evolutionary behaviour of the end-to-end approach to spoofing detection and then an assessment of performance in terms of the EER. Also discussed here is the behaviour of the gate.

5.1. Evolutionary behaviour

An illustration of the evolutionary behaviour of the end-to-end approach to spoofing detection is illustrated in Fig. 6. Two profiles show the evolution in EOC$_m$ (top blue profile) and the number of network node connections (green profile) of the EOC-fittest. The lower magenta profile shows the EER for the champion of each generation (the *generation champions*) estimated using training/development The single network selected for the testing/evaluation is that which produces the lowest EER for the training/development data (orange dot). This network is designated as the *grand champion* network.

[6] http://www.asvspoof.org/data2017/baseline_CM.zip

Table 2: *End-to-end spoofing detection performance for the ASVspoof 2017 database version 2 and extended protocol.*

	Train+Dev	**Eval**
Baseline	0.14%	23.4%
AUROC	20.9%	28.2%
AUROC$_m$	27.4%	24.2%
EOC	20.3%	19.2%
EOC$_m$	18.7%	18.2%

The fitness is seen to increase gradually as the end-to-end approach to anti-spoofing learns to discriminate between bona fide and spoofed speech, gradually increasing network complexity as it proceeds. Improvements in fitness are largely accompanied by decreases in EER. After approximately 350 iterations, the EER seems to converge, with the best performing network being that from the 484th generation and having 198 connections.

5.2. Spoofing detection performance

Results are presented in Table 2 for the baseline systems and the for the end-to-end system with AUROC and EOC fitness functions, with and without mini-batching (denoted by subscript m). Results for the EOC fitness function are either similar to or better than those for the AUROC fitness function. Mini-batching appears to offer inconsistent results for the AUROC fitness function; performance degrades for train+dev. but improves for evaluation. For the EOC fitness function, improvements are consistent across the two sets.

Of particular interest is the stability or generalisation achieved by the end-to-end system. Performance for the baseline system is seen to degrade substantially between the two sets (train+dev and evaluation). In contrast, the best results achieved with the end-to-end approach using the EOC fitness function and mini-batch training is not only substantially better, but also consistent across the two disjoint data sets (18%).

5.3. Gate operation

The gate acts to identify salient information in the network output. This is a form of an attention mechanism. As such, it is of interest to investigate its behaviour. Even so, the gate operates on the *output* stream rather than the *input* stream. Coupled with the recurrent nature of the network which maps inputs to outputs, this impedes a straightforward interpretation of its behaviour; it is difficult to interpret gate behaviour at the output with respect to the acoustic stream at the input.

Our investigations thus far show that the gate generates a somewhat periodic signal during both speech and non-speech intervals. This would indicate that information during both are of use to the detection of replay spoofing attacks. Further analysis would involve a deeper examination of how to link gate behaviour at the output to information at the input. This study is left for future work.

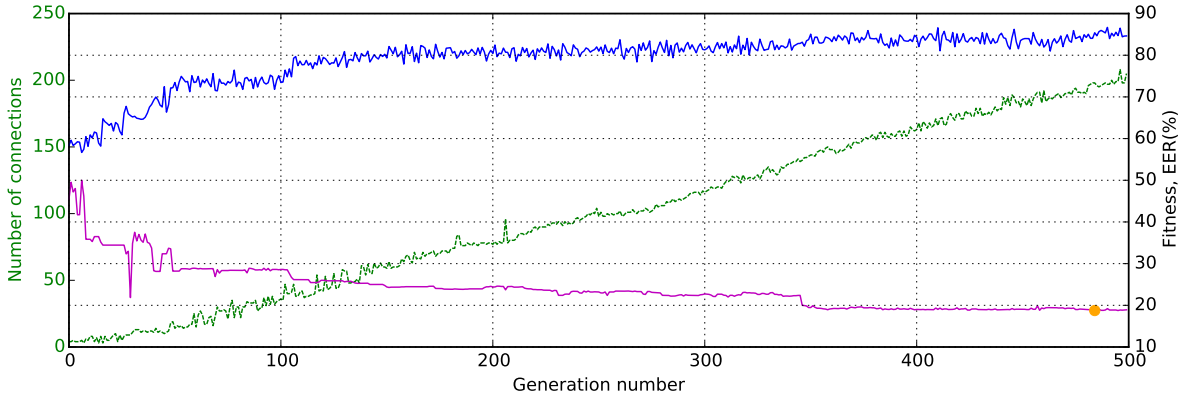


Figure 6: *Evolution of 500 generations with an EOC fitness function with mini-batch training. The upper blue profile shows the EOC-derived fitness of the fittest network in each generation. The highest fitness is obtained in generation 490. The green profile is the complexity (number of connections) in each network. The lower magenta profile is the EER of generation champions estimated using pooled training and development data. It reaches a minimum value in generation 484 (marked by an orange dot). This is the grand champion network that is chosen for testing on the evaluation set.*

6. Conclusions and future work

This paper reports a truly end-to-end approach to the problem of spoofing detection. End-to-end techniques that avoid a reliance upon hand-crafted features are assumed to offer better potential for spoofing detection, and especially generalisation when the cues indicative of spoofing can vary considerably and are largely unpredictable in practice. The paper show how the neuroevolution of augmenting topologies can be applied successfully to this task. Critical to performance is the proposed progress-rewarding fitness function which steers the evolutionary process progressively towards the reliable classification of a diminishing number of difficult trials. Coupled with a mini-batch training procedure, this particular quality of the proposed solution preserves generalisation.

Results for the ASVspoof 2017 Version 2.0 database show improvements to both generalisation and raw performance. Equal error rates for the end-to-end approach represent a 22% relative reduction compared to the baseline system. A particularly appealing feature of the end-to-end approach is the gate, which acts as a form of in-built attention mechanism which serves to distinguish the most reliable information in the network output. This aspect of the end-to-end solution requires further investigation in order to interpret its behaviour with respect to information present in the acoustic input. The findings of such a study, while left for further work, will help to determine precisely what information helps most to differentiate between bona fide and replayed speech.

7. References

[1] 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, 2010.

[2] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] K.A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *IEEE signal processing society speech and language technical committee newsletter*, February 2013.

[4] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *Proc. INTERSPEECH*, 2013, pp. 925–929.

[5] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, and A.-M. Laukkanen, "Automatic versus human speaker verification: The case of voice mimicry," *Speech Communication*, vol. 72, pp. 13–31, 2015.

[6] 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, 2015.

[7] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilci, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.

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

[9] F. Alegre, A. Janicki, and N. Evans, "Re-assessing the threat of replay spoofing attacks against automatic speaker verification," in *International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sept 2014, pp. 1–6.

[10] M. Grzywacz J. Gaka and R. Samborski, "Playback attack detection for text-dependent speaker verification over telephone channels," *Speech Communication*, vol. 67, pp. 143 – 153, 2015.

[11] R. Font, J. M. Espn, and M. J. Cano, "Experimental analysis of features for replay attack detection results on the ASVspoof 2017 challenge," in *Proc. INTERSPEECH*, 2017, pp. 7–11.

[12] H. A. Patil, M. R. Kamble, T. B. Patel, and M. H. Soni, "Novel variable length teager energy separation based instantaneous frequency features for replay detection," in *Proc. INTERSPEECH*, 2017, pp. 12–16.

[13] W. Cai, D. Cai, W. Liu, G. Li, and M. Li, "Countermeasures for automatic speaker verification replay spoofing attack : On data augmentation, feature representation,

classification and fusion," in *Proc. INTERSPEECH*, 2017, pp. 17–21.

[14] S. Jelil, R. K. Das, S. R. M. Prasanna, and R. Sinha, "Spoof detection using source, instantaneous frequency and cepstral features," in *Proc. INTERSPEECH*, 2017, pp. 22–26.

[15] M. Witkowski, S. Kacprzak, P. elasko, K. Kowalczyk, and J. Gaka, "Audio replay attack detection using high-frequency features," in *Proc. INTERSPEECH*, 2017, pp. 27–31.

[16] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," in *Proc. INTERSPEECH*, 2017, pp. 82–86.

[17] L. Li, Y. Chen, D. Wang, and T. F. Zheng, "A study on replay attack and anti-spoofing for automatic speaker verification," in *Proc. INTERSPEECH*, 2017, pp. 92–96.

[18] K. R. Alluri, S. Achanta, S. R. Kadiri, S. V. Gangashetty, and A. K. Vuppala, "SFF anti-spoofer: IIIT-H submission for automatic speaker verification spoofing and countermeasures challenge 2017," in *Proc. INTERSPEECH*, 2017, pp. 107–111.

[19] X. Wang, Y. Xiao, and X. Zhu, "Feature selection based on cqccs for automatic speaker verification spoofing," in *Proc. INTERSPEECH*, 2017, pp. 32–36.

[20] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatic spoofing detection," in *Proc. INTERSPEECH*, 2017, pp. 102–106.

[21] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using dnn for channel discrimination," in *Proc. INTERSPEECH*, 2017, pp. 97–101.

[22] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep Speaker Feature Learning for Text-independent Speaker Verification," *arXiv:1705.03670 [cs]*, May 2017.

[23] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection," in *IEEE IAPR International Joint Conference on Biometrics (IJCB)*, 2017.

[24] G. Valenti, A. Daniel, and N. Evans, "End-to-end automatic speaker verification with evolving recurrent neural networks," in *Speaker Odyssey 2018*.

[25] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[26] N. J. Radcliffe, "Genetic set recombination and its application to neural network topology optimisation," in *Neural Computing and Applications*, 1993, number 1, pp. 67–90.

[27] B. Allen and P. Faloutsos, "Complex networks of simple neurons for bipedal locomotion," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. 2009, pp. 4457–4462, IEEE.

[28] M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone, "A neuroevolution approach to general atari game playing," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 4, pp. 355–366, 2014.

[29] A. Daniel, "Evolving recurrent neural networks that process and classify raw audio in a streaming fashion," in *Proc. INTERSPEECH*, 2017.

[30] I. Jordal, "Evolving artificial neural networks for cross-adaptive audio effects," M.S. thesis, NTNU, 2017.

[31] C. Kroos and M. Plumbley, "Neuroevolution for sound event detection in real life audio: A pilot study," *Detection and Classification of Acoustic Scenes and Events (DCASE 2017) Proceedings*, 2017.

[32] C. Olah and S. Carter, "Attention and augmented recurrent neural networks," in *Distill*, 2016.

[33] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[34] T. Fawcett, "An introduction to ROC analysis," in *Pattern Recognition Letters*, 2006, number 27, pp. 861–874.

[35] J. A. C. Weideman, "Numerical integration of periodic functions: a few examples," in *The American Mathematical Monthly*, 2002, number 109, pp. 21–36.

[36] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[37] T. Kinnunen, M. Sahidullah., H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. Aik Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Proc. of INTERSPEECH*, 2017, pp. 2–6.

[38] J. Villalba and E. Lleida, *Biometrics and ID Management: COST 2011 European Workshop, BioID 2011, Brandenburg (Havel), Germany, March 8-10, 2011. Proceedings*, chapter Detecting Replay Attacks from Far-Field Recordings on Speaker Verification Systems, pp. 274–285, 2011.

[39] Z. Wu, S. Gao, E.S. Chng, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA*, 2014, pp. 1–5.

[40] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc. Odysssey*, Bilbao, Spain, 2016, pp. 283–290.

[41] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516 – 535, 2017.

[42] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, 2000.

[43] Douglas A. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*, 2009.

Bibliography

[1] J. Gonzalez-Rodriguez. Evaluating Automatic Speaker Recognition systems: An overview of the NIST Speaker Recognition Evaluations (1996-2014). *Loquens*, 1(1), June 2014. 3, 13, 15, 17, 23

[2] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, January 2010. 3, 12, 13, 15, 23

[3] L. P. Heck and D. Genoud. Integrating speaker and speech recognizers: Automatic identity claim capture for speaker verification. In *Odyssey Speaker and Language Recognition Workshop*, 2001. 3, 17, 31

[4] G. Valenti, A. Daniel, and N. Evans. A Simplified 2-Layer Text-Dependent Speaker Authentication System. In *143th Audio Engineering Society Convention*. Audio Engineering Society, 2017. 5, 30

[5] G. Valenti, A. Daniel, and N. Evans. On the Influence of Text Content on Pass-Phrase Strength for Short-Duration Text-Dependent Automatic Speaker Authentication. In *INTERSPEECH*, pages 3623–3627, 2016. 6, 39, 46

[6] G. Valenti, A. Daniel, and N. Evans. Spoken pass-phrase suitability determination, Patent US 2018/0060557 A1, 2016. http://www.eurecom.fr/publication/5503. 6, 39, 46, 93

[7] C. Kroos and M. Plumbley. Neuroevolution for sound event detection in real life audio: A pilot study. *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017. 6

[8] G. Valenti, A. Daniel, and N. Evans. End-to-end automatic speaker verification with evolving recurrent neural networks. In *Odyssey Speaker and Language Recognition Workshop*, 2018. 6, 54, 55, 57, 69, 72, 81

[9] G. Valenti, H. Delgado, M. Todisco, N. Evans, and L. Pilati. An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks. In *Odyssey Speaker and Language Recognition Workshop*, 2018. 6, 85

[10] Lee et al. The I4U submission to the 2016 NIST speaker recognition evaluation. In *NIST SRE 2016 Workshop*, 2016. 9, 74, 75, 98

[11] J. H. L. Hansen and T. Hasan. Speaker recognition by machines and humans: a tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015. 11, 13, 14, 15, 17, 19

[12] J.-F. Bonastre et al. Person Authentication by Voice: A Need for Caution. In *EUROSPEECH*, pages 33–36, 2003. 12

[13] F. Nolan. *The Phonetic Bases of Speaker Recognition*. Cambridge University Press, 1983. 12

[14] X. Zhou, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, and S. Shamma. Linear versus mel frequency cepstral coefficients for speaker recognition. In *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 559–564, Dec 2011. 13

[15] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980. 13

[16] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993. 14

[17] D. A. Reynolds and R. C. Rose et al. Robust text-independent speaker identification using Gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995. 15

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Ser. B*, 39(1):1–38, 1977. 15

[19] D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *EUROSPEECH*, 1997. 16, 17

[20] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3):19–41, January 2000. 17, 31, 69

[21] C.-H. Lee and J.-L. Gauvain. Speaker adaptation based on MAP estimation of HMM parameters. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 558–561. IEEE, 1993. 17

[22] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 17

[23] L. J. Rodríguez and I. Torres. Comparative study of the baum-welch and viterbi training algorithms applied to read and spontaneous speech recognition. In *Pattern Recognition and Image Analysis*, pages 847–857. Springer, 2003. 18

[24] A. Larcher, J.-F. Bonastre, and J. Mason. Reinforced temporal structure information for embedded utterance-based speaker recognition. In *INTERSPEECH*, pages 371–374, 2008. 18, 20, 31

[25] T. Kato and T. Shimizu. Improved speaker, verification over the cellular phone network using phoneme-balanced and digit-sequence-preserving connected digit patterns. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, April 2003. 18

[26] M. F. BenZeghiba and H. Bourlard. User-customized password speaker verification using multiple reference and background models. *Speech Communication*, 48(9):1200–1213, September 2006. 19

[27] R. Kuhn, P. Nguyen, J. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K.Field, and M. Contolini. Eigenvoices for speaker adaptation. In *Spoken Language Processing*, pages 1774–1777, 1998. 19

[28] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(5):273–297, 1995. 19

[29] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters*, 13(5):308–311, 2006. 19

[30] P. Kenny. Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms. *CRIM*, pages 18–29, 2005. 19

[31] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Support vector machines using GMM supervectors for speaker verification. *IEEE transactions on Audio, Speech, Language processing*, 19(4):788–798, 2011. 19

[32] D. Garcia-Romero and C. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *INTERSPEECH*, pages 249–252, 2011. 19

[33] A. Larcher, K. A. Lee, B. Ma, and H. Li. Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7673–7677. IEEE, 2013. 19

[34] P. Kenny, T. Stafylakis, P. Ouellet, Jahangirr Md. Alam, and P. Dumouchel. PLDA for speaker verification with utterances of arbitrary duration. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7649–7653. IEEE, 2013. 19

[35] A. Larcher, K. A. Lee, B. Ma, and H. Li. Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication*, 60:56–77, 2014. 19, 20, 26, 29, 32, 35, 42, 45

[36] K. A. Lee, A. Larcher, H. Thai, B. Ma, and H. Li. Joint Application of Speech and Speaker Recognition for Automation and Security in Smart Home. In *INTERSPEECH*, pages 3317–3318, 2011. 20

[37] A. Larcher, K.A. Lee, B. Ma, and H. Li. RSR2015: Database for text-dependent speaker verification using multiple pass-phrases. In *INTERSPEECH*, pages 1580–1583, 2012. 20, 26, 27, 30, 31, 35, 37, 42, 70, 92

[38] C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006. 20

[39] N. Brümmer and E. de Villiers. The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF. *arXiv:1304.2865*, 2013. 23

[40] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *International Conference on Spoken Language Processing (ICSLP)*, 1998. 23

[41] P. L. S. Martınez, B. Fauve, A. Larcher, and J. S. D. Mason. Speaker Verification Performance with Constrained Durations. In *International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2014. 24

[42] P. Kenny, N. Dehak, P. Ouellet, V. Gupta, and P. Dumouchel. Development of the primary CRIM system for the NIST 2008 speaker recognition evaluation. In *INTERSPEECH*, pages 1401–1404, 2008. 25, 29

[43] B. Fauve, N. Evans, and J.Mason. Improving the performance of text-independent short duration SVM-and GMM-based speaker verification. In *Odyssey Speaker and Language Recognition Workshop*, pages 18–25, 2008. 25, 29, 39

[44] A. Poddar, M. Sahidullah, and G. Saha. Performance comparison of speaker recognition systems in presence of duration variability. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2015. 25, 29

[45] A. Larcher, K. A. Lee, B. Ma, and H. Li. Imposture classification for text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014. 26

[46] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, et al. The RedDots data collection for speaker recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 26, 70

[47] K. A. Lee, B. Ma, and H. Li. Speaker verification makes its debut in smartphone, IEEE SLTC Newsletter, February 2013. 29

[48] H. Aronowitz. Voice Biometrics for User Authentication. In *Afeka-AVIOS Speech Processing Conference 2012*, 2012. 29

[49] M. Sahidullah and T. Kinnunen. Local spectral variability features for speaker verification. *Digital Signal Processing*, 50:1–11, March 2016. 29

[50] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel. I-Vector/PLDA variants for text-dependent speaker recognition. *CRIM Technical Report*, 2013. 29

[51] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel. Text-dependent speaker recognition using PLDA with uncertainty propagation. In *INTERSPEECH*, pages 3651–3655, 2013. 29, 31

[52] P. Kenny, T. Stafylakis, P. Ouellet, and Md J. Alam. JFA-based front ends for speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1705–1709. IEEE, 2014. 29

[53] P. Kenny, T. Stafylakis, J. Alam, and M. Kockmann. JFA modeling with left-to-right structure and a new backend for text-dependent speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4689–4693. IEEE, 2015. 29

[54] P. Kenny, T. Stafylakis, J. Alam, P. Ouellet, and M. Kockmann. Joint factor analysis for text-dependent speaker verification. In *Odyssey Speaker and Language Recognition Workshop*, pages 1705–1709, 2014. 29

[55] T. Stafylakis, P. Kenny, M. J. Alam, and M. Kockmann. Speaker and channel factors in text-dependent speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24:65–78, 2016. 29

[56] F. Bimbot, J.-F. Bonastre, and C. Fredouille et al. A tutorial on text-independent speaker verification. *EURASIP journal on applied signal processing*, 2004:430–451, 2004. 31

[57] A. Larcher, P.-M. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J.-F. Bonastre. I-vectors in the context of phonetically-constrained short utterances for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4773–4776. IEEE, 2012. 31

[58] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan. Further optimisations of constant Q cepstral processing for integrated utterance and text-dependent speaker verification. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 179–185. IEEE, December 2016. 31, 74

[59] A. Larcher, K. A. Lee, P. L. S. Martınez, T. H. Nguyen, B. Ma, and H. Li. Extended RSR2015 for text-dependent speaker verification over VHF channel. In *INTERSPEECH*, 2014. 31, 35, 36, 37, 45, 93, 100

[60] G. Soldi, S. Bozonnet, F. Alegre, C. Beaugeant, and N. Evans. Short-duration speaker modelling with phone adaptive training. In *Odyssey Speaker and Language Recognition Workshop*, 2014. 39

[61] J. Kahn, S. Rossato, and J.-F. Bonastre. Beyond doddington menagerie, a first step towards. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4534–4537, 2010. 39

[62] J. Kahn, N. Audibert, J.-F. Bonastre, and S. Rossato. Inter and intraspeaker variability in french: an analysis of oral vowels and its implication for automatic speaker verification. In *International Congress of Phonetic Sciences (ICPhS)*, pages 1002–1005, 2011. 39

[63] K. Amino, T. Sugawara, and T. Arai. Idiosyncrasy of nasal sounds in human speaker identification and their acoustic properties. *Acoustical science and technology*, 27(4):233–235, 2006. 40, 46

[64] D. Howell. *Statistical methods for psychology, 7th edition*. Cengage Learning, 2010. 43

[65] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015. 47, 48, 50, 53

[66] S. S. Tirumala and S. R. Shahamiri. A review on Deep Learning approaches in Speaker Identification. In *8th International Conference on Signal Processing Systems - ICSPS*, pages 142–147. ACM Press, 2016. 47

[67] B. C. Csáji. *Approximation with Artificial Neural Networks*. Faculty of Sciences, Eötvös Loránd University, 2001. 47

[68] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 47

[69] M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. 48

[70] L. J. Ba and R. Caruana. Do Deep Nets Really Need to be Deep? *arXiv:1312.6184 [cs]*, December 2013. arXiv: 1312.6184. 48

[71] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *arXiv:1605.07146 [cs]*, May 2016. arXiv: 1605.07146. 48

[72] G. E. Hinton, S. Osindero, and Y.-W. Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, July 2006. 48

[73] J. Martens. Deep learning via Hessian-free optimization. In *27th International Conference on Machine Learning ICML*, pages 735–742, 2010. 48

[74] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *30th International Conference on Machine Learning*, pages 1139–1147, 2013. 48

[75] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014. 48

[76] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385. 48, 55

[77] G. E. Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009. revision #91189. 49

[78] D. Cireşan, U. Meier, and J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. *arXiv:1202.2745 [cs]*, February 2012. 50

[79] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. 51

[80] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. 52

[81] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056. IEEE, 2014. 52, 53

[82] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang. Deep speaker feature learning for text-independent speaker verification. In *INTERSPEECH*, pages 1542–1546, 2017. 52

[83] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012. 52

[84] Y. Zhang, E. Chuangsuwanich, and J. R. Glass. Extracting deep neural network bottleneck features using low-rank matrix factorization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 185–189, 2014. 52

[85] S. Dey, S. Madikeri, M. Ferras, , and P. Motlicek. Deep neural network based posteriors for text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5050–5054, 2016. 52

[86] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1695–1699, 2014. 52

[87] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *INTERSPEECH*, pages 999–1003, 2017. 52

[88] F. Richardson, D. Reynolds, and N. Dehak. Deep Neural Network Approaches to Speaker and Language Recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675, October 2015. 53

[89] H.-S. Lee, Y.-D. Lu, C.-C. Hsu, Y. Tsao, H.-M. Wang., and S.-K. Jeng. Discriminative autoencoders for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 53

[90] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer. End-to-end text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016. 53, 54

[91] A. Miguel, J. Llombart, A. Ortega, and E. Lleida. Tied Hidden Factors in Neural Networks for End-to-End Speaker Recognition. In *INTERSPEECH*, pages 2819–2823, 2017. 53

[92] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong. End-to-End Attention based Text-Dependent Speaker Verification. *arXiv:1701.00562*, January 2017. 53

[93] A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, Jan 2012. 53

[94] Y. Chen, I. Lopez-Moreno, T. N. Sainath, M. Visontai, R. Alvarez, and C. Parada. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *INTERSPEECH*, pages 1136–1140, 2015. 53

[95] H. Sak, A. Senior, and F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint:1402.1128*, 2014. 53

[96] E. Godoy, N. Malyska, and T. F. Quatieri. Estimating Lower Vocal Tract Features with Closed-Open Phase Spectral Analyses. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 54

[97] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5200–5204. IEEE, 2016. 54

[98] H. Muckenhirn, M. Magimai-Doss, and S. Marcel. End-to-End Convolutional Neural Network-based Voice Presentation Attack Detection. In *IEEE IAPR International Joint Conference on Biometrics (IJCB)*, 2017. 54

[99] Z. Tüske, P. Golik, R. Schlüter, and H. Ney. Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *INTERSPEECH*, pages 890–894, 2014. 54

[100] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *INTERSPEECH*, pages 1–5, 2015. 54

[101] Y. Hoshen, R. J. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4624–4628. IEEE, 2015. 54

[102] K. Tokuday and H. Zen. Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4215–4219. IEEE, 2015. 54

[103] A. Van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint:1609.03499*, 2016. 54

[104] H. Muckenhirn. Towards directly modeling raw speech signal for speaker verification using CNNs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. 54, 57, 64

[105] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yu. A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. 54, 57, 64

[106] M. Ravanelli and Y. Bengio. Speaker Recognition from Raw Waveform with SincNet. In *INTERSPEECH*, July 2018. 54, 57, 64

[107] Y. Chen, I. Lopez-Moreno, T. N. Sainat, M. Visontai, R. Alvarez, and C. Parada. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *INTERSPEECH*, pages 999–1003, 2015. 54, 72

[108] B. Baker, O. Gupta, N. Naik, and R. Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016. 55

[109] B. T. Zhang and H. Muhlenbein. Evolving optimal neural networks using genetic algorithms with occam's razor. *Complex Systems*, (7):199–220, 1993. 55

[110] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002. 58

[111] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. 58

[112] D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008. 58

[113] Z. Chen, Z. Xie, W. Zhang, and X. Xu. Resnet and model fusion for automatic spoofing detection. In *INTERSPEECH*, pages 102–106, 2017. 59, 80

[114] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002. 59, 60, 61, 63

[115] N. J. Radcliffe. Genetic set recombination and its application to neural network topology optimisation. In *Neural Computing and Applications*, number 1, pages 67–90, 1993. 60

[116] J. Lehman and K. O. Stanley. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *ALIFE*, pages 329–336, 2008. 62, 66

[117] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *12th annual conference on Genetic and evolutionary computation*, pages 103–110. ACM, 2010. 62

[118] B. Allen and P. Faloutsos. Complex networks of simple neurons for bipedal locomotion. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4457–4462. IEEE, 2009. 62

[119] M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone. A neuroevolution approach to general atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(4):355–366, 2014. 62

[120] D. V. Vargas and J. Murata. Spectrum-Diverse Neuroevolution With Unified Neural Models. *IEEE Transactions on Neural Networks and Learning Systems*, 28(8):1759–1773, August 2017. 62

[121] I. Jordal. Evolving artificial neural networks for cross-adaptive audio effects. Master's thesis, NTNU, 2017. 62

[122] C. Kroos and M. Plumbley. Neuroevolution for sound event detection in real life audio: A pilot study. In *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017. 62

[123] A. Daniel. Evolving recurrent neural networks that process and classify raw audio in a streaming fashion. In *Proc. INTERSPEECH*, 2017. 64, 65, 81

[124] C. Olah and S. Carter. Attention and augmented recurrent neural networks. In *Distill*, 2016. 64

[125] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015. 64

[126] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, (27):861–874, 2006. 65

[127] J. A. C. Weideman. Numerical integration of periodic functions: a few examples. *The American Mathematical Monthly*, (109):21–36, 2002. 66

[128] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Springer, 2010. 66

[129] T. N. Sainath and C. Parada. Convolutional Neural Networks for Small-Footprint Keyword Spotting. In *INTERSPEECH*, page 5, 2015. 69

[130] P. Nakkiran, R. Alvarez, R. Prabhavalkar, and C. Parada. Compressing deep neural networks using a rank-constrained topology. In *INTERSPEECH*, pages 1473–1477, 2015. 69

[131] S. O. Sadjadi, T. Kheyrkhah, A. Tong, and C. Greenberg et al. The 2016 NIST Speaker Recognition Evaluation. In *INTERSPEECH*, pages 1353–1357. ISCA, 2017. 70

[132] J. S. Garofolo, L. F. Lamel, W. M. Fisher, and J. G. Fiscus et al. TIMIT acoustic-phonetic continuous speech corpus. LDC93S1, Philadelphia: Linguistic Data Consortium, Web Download, 1993. 70

[133] ITU-T Recommendation P.56: Objective measurement of active speech level, 2011. http://www.itu.int/rec/T-REC-P.56-201112-I/en. 70

[134] N. Evans, T. Kinnunen, and J. Yamagishi. Spoofing and countermeasures for automatic speaker verification. In *INTERSPEECH*, pages 925–929, 2013. 80

[135] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, and A.-M. Laukkanen. Automatic versus human speaker verification: The case of voice mimicry. *Speech Communication*, 72:13–31, 2015. 80

[136] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130 – 153, 2015. 80

[137] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilci, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado. ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604, 2017. 80

[138] J. Villalba and E. Lleida. Preventing replay attacks on speaker verification systems. In *2011 Carnahan Conference on Security Technology*, pages 1–8, Oct 2011. 80

[139] F. Alegre, A. Janicki, and N. Evans. Re-assessing the threat of replay spoofing attacks against automatic speaker verification. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2014. 80, 85

[140] M. Grzywacz J. Gałka and R. Samborski. Playback attack detection for text-dependent speaker verification over telephone channels. *Speech Communication*, 67:143 – 153, 2015. 80

[141] R. Font, J. M. Espín, and M. J. Cano. Experimental analysis of features for replay attack detection — results on the ASVspoof 2017 challenge. In *INTERSPEECH*, pages 7–11, 2017. 80

[142] H. A. Patil, M. R. Kamble, T. B. Patel, and M. H. Soni. Novel variable length teager energy separation based instantaneous frequency features for replay detection. In *INTERSPEECH*, pages 12–16, 2017. 80

[143] W. Cai, D. Cai, W. Liu, G. Li, and M. Li. Countermeasures for automatic speaker verification replay spoofing attack : On data augmentation, feature representation, classification and fusion. In *INTERSPEECH*, pages 17–21, 2017. 80

[144] S. Jelil, R. K. Das, S. R. M. Prasanna, and R. Sinha. Spoof detection using source, instantaneous frequency and cepstral features. In *INTERSPEECH*, pages 22–26, 2017. 80

[145] M. Witkowski, S. Kacprzak, P. Żelasko, K. Kowalczyk, and J. Gałka. Audio replay attack detection using high-frequency features. In *INTERSPEECH*, pages 27–31, 2017. 80

[146] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin. Audio replay attack detection with deep learning frameworks. In *INTERSPEECH*, pages 82–86, 2017. 80

[147] L. Li, Y. Chen, D. Wang, and T. F. Zheng. A study on replay attack and anti-spoofing for automatic speaker verification. In *INTERSPEECH*, pages 92–96, 2017. 80

[148] K. R. Alluri, S. Achanta, S. R. Kadiri, S. V. Gangashetty, and A. K. Vuppala. SFF anti-spoofer: IIIT-H submission for automatic speaker verification spoofing and countermeasures challenge 2017. In *INTERSPEECH*, pages 107–111, 2017. 80

[149] X. Wang, Y. Xiao, and X. Zhu. Feature selection based on CQCCs for automatic speaker verification spoofing. In *INTERSPEECH*, pages 32–36, 2017. 80

[150] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland. Replay attack detection using DNN for channel discrimination. In *Proc. INTERSPEECH*, pages 97–101, 2017. 80

[151] T. Kinnunen, M. Sahidullah., H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In *INTERSPEECH*, pages 2–6, 2017. 84

[152] J. Villalba and E. Lleida. *Biometrics and ID Management: COST 2101 European Workshop, BioID 2011*, chapter Detecting Replay Attacks from Far-Field Recordings on Speaker Verification Systems, pages 274–285. 2011. 85

[153] Z. Wu, S. Gao, E.S. Chng, and H. Li. A study on replay attack and anti-spoofing for text-dependent speaker verification. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA*, pages 1–5, 2014. 85

[154] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi. ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements. In *Odyssey Speaker and Language Recognition Workshop*, pages 296–303. ISCA, June 2018. 85

[155] M. Todisco, H. Delgado, and N. Evans. A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients. In *Odyssey Speaker and Language Recognition Workshop*, pages 283–290, Bilbao, Spain, 2016. 85

[156] M. Todisco, H. Delgado, and N. Evans. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516 – 535, 2017. 85

[157] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 2000. 85

[158] D. A. Reynolds. Gaussian mixture models. In *Encyclopedia of Biometrics*, 2009. 85

[159] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9:2579–2605, 2008. 100