SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS
DEPUIS 1257

EURECOM
Sophia Antipolis

End-to-End Modeling for Speech Spoofing and Deepfake Detection

Dissertation

submitted to

Sorbonne Université

in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Author:
Hemlata Tak

Defended on the
23rd of May 2023
before a committee composed of:

Thesis advisor	**Prof. Nicholas EVANS**, EURECOM, France
Reviewers	**Prof. Anthony Larcher**, LIUM—Le Mans Université, France
	Dr. Emmanouil Benetos, Queen Mary University of London, UK
Examiners	**Prof. Davide Balzarotti**, EURECOM, France
	Dr. Jennifer Williams, University of Southampton, UK

Acknowledgements

I would like to express my gratitude to my adviser Prof. Nicholas Evans for giving me the opportunity to join his excellent team Audio Security and Privacy group at EURECOM. Under Nick's unwavering tutelage, I have been able to hone my technical skills and grow both professionally and personally during my Ph.D. I'm also deeply grateful to my co-supervisor Dr. Massimiliano Todisco, whose guidance, constant support and meticulousness have motivated me throughout my PhD journey.

Furthermore, I would like to extend my gratitude to all the members of my thesis jury committee for generously offering their time, for their insightful feedback and suggestions: Prof. Anthony Larcher, Dr. Emmanouil Benetos, Porf. Davide Balzarotti, and Dr. Jennifer Williams. I also would like to acknowledge the invaluable support and guidance of all my collaborators. A special thanks goes to Prof. Tomi Kinnunen, Prof. Junichi Yamagishi, Dr. Kong Aik Lee, Dr. Jee-weon Jung and Dr. Xin Wang.

At Eurecom, I have been fortunate enough to be surrounded by excellent colleagues and friends who have contributed to developing a pleasant and enjoyable working environment. From our group: Wanying, Michele and Oubaida, with whom I have shared many cherishable moments of joy. From the other colleagues that are or were at EURECOM: Pepe, Khawla, Pramod, Leela, Madhu, Rakesh, Adeel, Rajeev. Friends from other universities: Neil Zhang, Hyejin and Xuechen. Thanks for the lovely years.

Finally, I would like to thank my family back in India, to my parents for their unconditional love, their never-ending sacrifices, and to my siblings for their constant support through love and prayers for good health during my PhD.

Antibes, May 2023 ***Hemlata Tak***

Abstract

Voice biometric systems are being used more and more in various applications, including banking, call-centres, airports, access control and forensics. These systems use automatic speaker verification technology for secure user authentication but are susceptible to spoofing attacks, also known as presentation attacks. Spoofing is now a growing concern in academia and industry. It is essential to mitigate the threat, especially in high security scenarios. Recent advances in artificial intelligence have greatly improved the capability of generating synthetic voices, making it even more challenging to distinguish between genuine and fake audio. There is hence a need for more robust, and efficient detection techniques. This thesis proposes novel detection algorithms which are designed to perform reliably in the face of the highest quality attacks.

The first contribution is a non-linear ensemble of sub-band classifiers each of which uses a classical Gaussian mixture model (GMM). Competitive results with such a traditional approach show that models which learn sub-band specific discriminative information can substantially outperform models trained on full-band signals. Given that deep neural networks are more powerful than GMMs and can perform both feature extraction and classification, the second contribution of this thesis is a RawNet2 model. It is an end-to-end approach to anti-spoofing and deepfake detection which automatically learns discriminative features directly from raw waveform inputs. Results show that RawNet2 performs reliably even in the face of previously unseen spoofing attacks. End-to-end modelling can be seen as a joint feature extraction and classification framework which streamlines the processes of training and evaluation.

The third contribution of this thesis includes the first use of graph neural networks (GNNs) with an attention mechanism to model the complex relationship between discriminative information present in spectral and temporal domains. We propose an end-to-end spectro-temporal graph attention network called RawGAT-ST. Like the RawNet2 model, it also operates directly upon raw waveform inputs. An attentive graph pooling layer is incorporated to identify and retain informative

nodes and to discard irrelevant ones, thereby reducing computation and also improving discrimination power. The RawGAT-ST model is further extended to an integrated spectro-temporal graph attention network, named AASIST which exploits the relationship between heterogeneous spectral and temporal graphs. The use of a heterogeneous graph attention network allows for the integration of different types of nodes/edges which contain different feature characteristics. GNN-based countermeasures leverage the inherent information in both domains concurrently, improving the detection of more sophisticated spoofing attacks, while also improving upon generalisation.

The final contributions relate to the development of a novel data augmentation technique and a self-supervised front-end which improves generalisation and domain-robustness under more practical conditions. Acquiring training data that is representative of spoofing attacks with near-boundless variability is impractical or even impossible. Nonetheless, the performance of spoofing countermeasures relies on the use of sufficiently representative training data. To address this issue, we propose a raw data augmentation technique called RawBoost. RawBoost improves spoofing detection reliability in the face of nuisance variation stemming from unknown encoding, and transmission conditions and from different microphones and amplifiers, and both linear and non-linear device-generated distortion, all of which characterise a logical access or telephony scenario. An alternative approach is to use a front-end in the form of readily available self-supervised, pre-trained speech models trained on large databases. The combination of a self-supervised front-end with RawBoost brings substantial improvements in performance for the ASVspoof 2021 logical access and deepfake databases.

The work reported in this thesis has redefined the state-of-the-art in anti-spoofing, with results for RawGAT-ST, AASIST and SSL-based countermeasure solutions all being the best reported at the time of publication, and with those for the self-supervised based countermeasure remaining the best reported to date.

Contents

List of Figures

List of Tables

List of abbreviations

ASV	Automatic Speaker Verification
CM	Countermeasure
CNN	Convolutional Neural Network
CQCC	Constant Q Cepstral Coefficient
CQT	Constant Q Transform
DA	Data Augmentation
DF	Deepfake
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
EER	Equal Error Rate
FFT	Fast Fourier Transform
GAT	Garph Attention Network
GCN	Graph Convolutional Network
GMM	Gaussian Mixture Model
GNN	Garph Neural Network
Gpool	Graph Pooling Layer
ISD	Impulsive Signal-Dependent
LA	Logical access
LFCC	Linear Frequency Cepstral Coefficient
PA	Physical access
SA	Self-Attentive Aggregation
SIA	Signal-Independent Additive
SSL	Self Supervised Learning
SVM	Support Vector Machine
t-DCF	tandem Detection Cost Function
TTS	Text to Speech Synthesis
VC	Voice Conversion

Chapter 1

Introduction

The focus of the work presented in this thesis is the design of generalisable countermeasures to secure voice biometric systems from spoofing attacks. These countermeasures are designed to be domain-robust and efficient, and to function reliably in real-world environments.

1.1 Biometric system vulnerabilities

Voice-based human-machine interfaces [1] are today widely used for commercial services, such as online banking and e-commerce. These systems use automatic speaker verification (ASV) technology [1–4] to verify the identity of a user before allowing them to interact with the service and to access sensitive information or resources. While ASV serves as a convenient and efficient approach for user authentication, like any biometric system, it can be vulnerable to spoofing attacks [5]. According to the ISO/IEC 30107-1 standards [5,6], a generic biometric system recognition (i.e., ASV) can be manipulated or attacked at various points, as shown in Figure 1.1. Spoofing, the focus in this thesis, is applied by an adversary at the points marked ① and ② in Figure 1.1. Since neither sensor level ① nor post-sensor level ② attacks need system-level access, they are more easily implemented than other forms of attack at different system levels ③-⑨ and are generally considered to pose the greatest threat [7]. In an ASV scenario sensor-level (microphone) attacks are assumed to be launched in a physical access (PA) scenario whereas post-sensor-level (post-microphone) attacks correspond to a logical access (LA) scenario [8]. In the former, the microphone is a part of the biometric system and within the control of the system designer. In LA scenarios, e.g. telephony, the microphone is not part of the biometric system and is, instead, under the control of the user. With telephony being a dominant ASV use case, the work in this thesis is concerned predominantly with post-sensor attacks and the LA scenario.

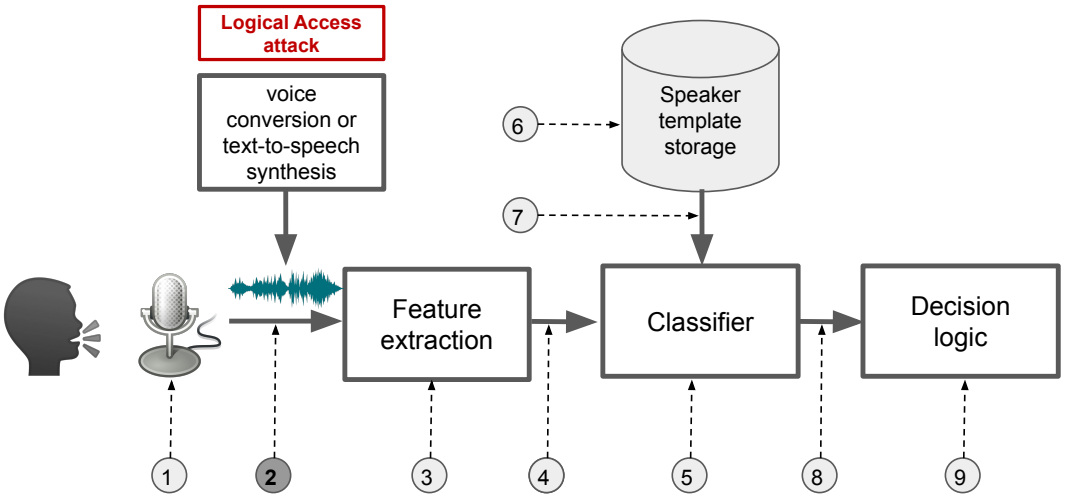


Figure 1.1: Possible presentation attacks point to speaker recognition system. 1: microphone point, 2: transmission point, 3: override feature extractor, 4: modify features, 5: override classifier, 6: modify speaker database, 7: modify biometric reference, 8: modify score and 9: override decision. Reproduced from [8].

1.2 Spoofing

ASV spoofing attacks [5], involve the manipulation of biometric recognition through various methods, such as voice conversion (VC) [9, 10], text-to-speech synthesis (TTS) [11], and replay attacks [12, 13], all of which can degrade performance. A growing number of studies have gauged the vulnerability of ASV systems to various forms of attack [14, 15]. For applications requiring medium to high-level security, the susceptibility to spoofing is a critical concern that cannot be ignored. Advanced attack algorithms can increasingly generate convincing fake utterances which can fool ASV systems. They can be extremely difficult to detect or distinguish from bona fide speech. A typical spoofing generation framework is illustrated to the left in Figure 1.2. The spoofing generation process utilises advanced TTS and VC techniques to create spoofed speech. They introduce processing artefacts that may be used by a countermeasure (CM) for spoofing detection.

TTS systems generate entirely artificial speech signals from text inputs, while VC systems operate on natural (human) speech. TTS-based attacks are highly effective, and many synthetic speech detectors have been designed to protect against TTS-based spoofing attacks [16]. Synthetic speech can be generated with para-

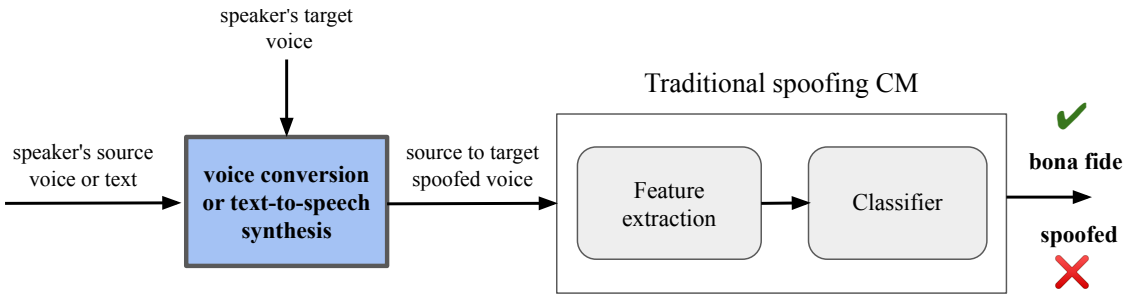


Figure 1.2: A traditional spoofing generation and detection framework.

metric synthesis frameworks [17], using the WORLD vocoder [18], WaveNet [19], and waveform concatenation methods [20] among others. Phase-based synthetic detectors were the state-of-the-art for synthetic speech detection [21, 22] at the time when the work presented in this thesis began. Today's more advanced end-to-end neural TTS system, e.g. Tacotron 2 [23] can generate synthetic speech with high naturalness and greater perceptual similarity to target speakers and is generally more challenging to detect. VC-based attacks convert the voice in a given utterance towards a target voice in order to deceive ASV systems and can be generated using VAE-based frameworks [24] with the WORLD vocoder, Wavenet vocoder [25], and with waveform and spectral filtering through classical overlap-add (OLA) techniques, among others. A more detailed treatment of TTS and VC spoofing attacks and CMs for automatic speaker verification, at the time when this work began, can be found in [8, 26–29].

The recent advancement in techniques to generate realistic audio content has left it difficult to distinguish between real and fake content. While they have useful applications in real life, they can also lead to serious threats related to security through so-called *Deepfakes*. In recent years, these threats have gained increasing attention [28, 30, 31], as demonstrated by the dangers of fake audio recordings in spreading misinformation, fraud, phishing and identity theft [32]. Adversaries are already using speech deepfakes to commit fraud [33]. Advances in synthesis and deepfake technology have made it easier to generate credible synthetic voice signals that can manipulate recognition systems [34]. Audio deepfakes are easily accessible by anyone using a computer device or a simple smartphone [35]. Spoofing attacks and deepfakes are now a growing concern. The protection of ASV systems from such threats has become essential, and has led to the development of spoofing CMs, also known as presentation attack detection (PAD) solutions in the ISO/IEC 30107 standard [5]. PAD solutions are usually considered as either artefact detection or liveness detection [36, 37]. Liveness detection normally requires additional sensors (e.g., airflow and throat sensors to

detect airflow and throat vibrations) which might not be available in the case of telephony scenarios. Most speaker recognition PAD solutions are based on artefact detection, which is the focus in this thesis.

The Automatic Speaker Verification Spoofing and Countermeasures (ASVspoof) initiative was established to promote the development of CMs/PADs to protect ASV systems from being spoofed. The role of a CM is to determine whether a speech utterance is bona fide speech or is, instead, spoofed. A traditional spoofing CM framework is illustrated to the right in Figure 1.2. It comprises a front-end (feature extraction), a back-end, and decision logic (classifier), where the final classification decision is made based upon the comparison of a classification score to a pre-defined detection threshold. Speech spoken by a human speaker is referred to as *bona fide* or genuine speech, whereas that generated either by machines or by replaying a recorded utterance is referred to as *spoofed*. A detailed summary of spoofing CM systems can be found in widely cited review articles [8, 29, 38, 39]. Voice spoof detection has now become a well-established research topic, bringing together researchers from various fields such as biometrics, machine learning, and speech processing.

The first special event on automatic speaker verification spoofing and countermeasures was held at INTERSPEECH in 2013 [14]. The ASVspoof community has since organised four biennial challenges to promote the development of spoofing CMs to protect ASV systems. These challenges have provided common datasets of bona fide and spoofed speech signals, baseline systems, and platforms for the evaluation of different CM solutions. Initially, the ASVspoof initiative established research in the same two scenarios introduced in Section 1.1, an LA scenario involving spoofing attacks generated by TTS and VC technologies, and a PA scenario, involving attacks produced by recording and replay devices in controlled, simulated and real scenarios. More recently, a new deepfake (DF) detection task was introduced and involves attacks generated from different sources and audio data which is compressed using algorithms typical of online and social media scenarios.

The first edition of the ASVspoof challenge series, held in 2015 [15], focused on developing CMs for the detection of LA attacks generated using TTS and VC technologies. The second edition of the challenge, held in 2017, shifted the focus to PA attacks, specifically replay spoofing attacks [40]. The third edition, ASVspoof 2019, was the first to combine both LA and PA scenarios, including TTS, VC, and replay spoofing attacks, in a single evaluation. The goal of the LA evaluation was to determine the extent to which more advanced speech synthesis

and voice conversion technology pose a threat to the reliability of ASV systems. The performance of a spoofing CM has a direct impact on the effectiveness of the ASV system. While CMs can improve security by rejecting spoofed trials, they can also lead to degraded usability by rejecting bona fide trials. As a result, it will impact on both reliability and usability. Accordingly, there is no assurance that a better-performing CM with lower a equal error rate (EER) will deliver more reliable ASV performance. Hence, an integrated approach to assessment is desirable and should measure the effect of spoofing and CMs upon the ASV system. To address this, the ASVspoof 2019 challenge introduced a new tandem evaluation method, using the minimum tandem Detection Cost Function (min t-DCF) as the default evaluation metric [40]. It replaced the EER used in previous editions.

Despite significant progress in spoofing detection reliability, generalisation and reliability remain a challenge in real-world conditions. This issue was addressed in the fourth edition, ASVspoof 2021 [41]. In the first three editions of the ASVspoof challenge series, bona fide and spoofed data were of high quality (clean audio) without the variation which could be expected in real-world conditions, e.g. variation stemming from transmission through telephony networks, encoding, and compression effects. CMs trained on such clean data may not generalise well to more practical scenarios, such as the telephony [42, 43]. The fourth edition of the challenge series, ASVspoof 2021, aimed to address this issue by promoting the development of CMs which improve generalisation and domain robustness in more realistic scenarios involving speech data transmission. The new DF detection task introduced for the ASVspoof 2021 challenge, aimed to improve CM robustness in the face of compressed speech across diverse domains. The DF database was generated from multiple source corpora, including the Voice Cloning toolkit (VCTK) database [44], the 2018 [45] and 2020 [46] Voice Conversion Challenge (VCC) databases. It contains spoofed utterances generated with over 100 diverse spoofing attack algorithms. This thesis focuses primarily on the LA and DF scenarios.

1.3 Thesis scope

Over the past few decades, research has led to the development of robust spoofing detection solutions and significant improvements in performance. The best performing CMs are often an ensemble of multiple systems. This trend was particularly evident in the ASVspoof 2019 challenge, for which a selection of detection error trade-off (DET) results are presented in Figure 1.3. The best ensemble sys-

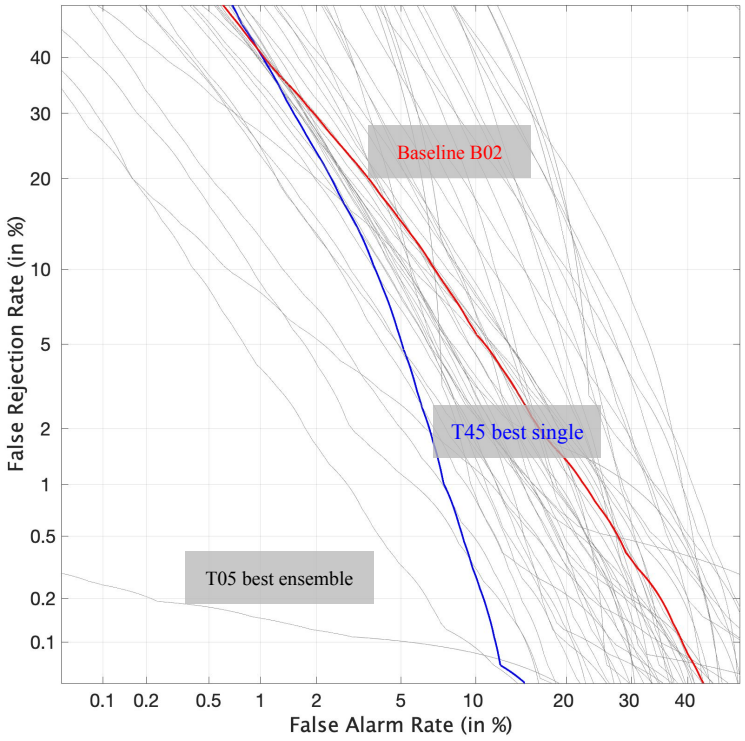


Figure 1.3: Countermeasures DET profiles for the ASVspoof 2019 LA challenge submissions. All grey profiles for ensemble systems and the best single system and baseline are highlighted in blue and red profile, respectively.

tem, T05[1], combined the outputs of seven complex deep neural network-based sub-models, as described in [47]. Due to the lack of a detailed system description and open source implementation, this level of performance has never been reproduced by others. From ASVspoof results, it is evident that min t-DCF results are dominated by performance for some worst-case attack algorithms. An adversary could exploit knowledge of such worst-case conditions to use only the most effective attack to better manipulate an ASV system. Therefore, it is important to improve not just pooled (across different spoofing attacks) performance, but also performance in these worst-case scenarios (most difficult-to-detect attacks). Generalisation to a wide range of attacks remains an open challenge. To improve model generalisation and push the limits of spoofing CM performance, the work reported in this thesis had the aim of designing more reliable, more generalisable and more efficient spoofing CMs. Ensemble-based solutions (e.g. T05) can be computationally expensive, making it difficult to deploy these models in real-world applications. There is a need for more computationally efficient CMs that can achieve better,

[1]Anonymous identifiers were used in order to protect team identities.

or similar levels of performance as existing complex ensemble-based solutions.

1.4 Thesis structure

The work presented in this thesis endeavors to develop reliable spoofing CMs for secure voice biometrics which generalise well across different realistic environments. We introduce new end-to-end deep neural network and graph neural network-based CMs, namely, RawNet2, RawGAT-ST and AASIST as well as a self-supervised learning-based solution, in addition to a novel data augmentation technique named RawBoost. We present an evaluation of the proposed methods and techniques performed using three widely used standard benchmarks, namely ASVspoof 2019 LA, and ASVspoof 2021 LA and DF. These are described in Chapter 2, which also presents a literature review of ASV vulnerabilities to spoofing as well as CMs, and the progress made in improving spoofing detection reliability through biennial ASVspoof challenges. Conclusions and future research directions are presented in Chapter 10. The following provides an overview of the structure and contributions in chapters 3-9.

Chapter 3

The work presented in Chapter 3 investigates the impact of sub-band modeling for ASV spoofing detection. We present an explainability study of constant-Q Cepstral Coefficients (CQCCs), one of the most popular spoofing CM front-ends when this work began. The goal of this work is to better understand and explain why the CQCC front-end is so effective at detecting some forms of spoofing attack but performs poorly in detecting others. Our findings from sub-band modeling confirm that different spoofing attacks exhibit artefacts at different frequencies, which can be better captured by specific front-ends. A proper understanding of these artefacts and where they are localised in the signal helps to design more reliable spoofing CMs, as shown later in the thesis.

The work presented in this chapter was published in:

- **Hemlata Tak**, Jose Patino, Andreas Nautsch, Nicholas Evans and Massimiliano Todisco, "**An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification**," in Proc. *The Speaker and Language Recognition Workshop*, Tokyo, Japan, Nov., 2020.

Chapter 4

Based on findings from Chapter 3, in Chapter 4 we first investigate whether spoofing attacks leave sub-band artefacts or cues that require specific spoofing CMs to detect. Second, we investigate whether non-linear fusion approaches offer the potential to combine the scores produced by an ensemble of different sub-band classifiers. Our results show that a simple sub-band modeling-based approach can achieve superior performance compared to more sophisticated ensemble solutions which rely on more complex and deeper neural network architectures.

The work presented in this chapter was published in:

- **Hemlata Tak**, Jose Patino, Andreas Nautsch, Nicholas Evans and Massimiliano Todisco, "**Spoofing Attack Detection using the Non-linear Fusion of Sub-band Classifiers**," in *Proc. INTERSPEECH*, Shanghai, China, October 2020.

This work shows that a non-linear ensemble of sub-band CMs, which learn sub-band specific discriminative information can substantially outperform models trained on full-band signals.

Chapter 5

Following the emphasis on hand-crafted features in Chapters 3 and 4, in Chapter 5 the focus shifts to the use of end-to-end modeling, where the front-end and back-end classifier are jointly optimised. The aim is to design an end-to-end deep neural network that can learn more generalisable and task-specific features directly from the raw waveform. Our hypothesis is that hand-crafted features do not offer the best potential for detecting unforeseen attacks because they rely too heavily on the characterisation of artefacts or cues corresponding to *known* attacks. These attack-specific artefacts may be insufficient for detecting previously unseen attacks, so a higher-level, more generalisable representation is needed to ensure robust performance. Introduced in Chapter 5 is the first successful application of RawNet2, an end-to-end deep neural network for spoofing and deepfake detection. It is used to learn representative features directly from raw waveform inputs in fully end-to-end fashion. Inspired by the importance of the spectro-temporal resolution in Chapter 3, we also investigate the use of different spectral and temporal resolutions in the front-end initialisation to capture artefacts more effectively. The proposed end-to-end approach achieves impressive results for a range of different spoofing attacks, and outperforms all prior approaches for the *worst-case* scenario (the most difficult-to-detect A17 attack) at the time of publication.

The work presented in this chapter was published in:

- **Hemlata Tak**, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans and Anthony Larcher, "**End-to-end anti-spoofing using RawNet2**," in *Proc. IEEE ICASSP*, Toronto, Ontario, Canada, June, 2021.

This work introduces a new end-to-end architecture, RawNet2 which operates directly upon raw waveform inputs to effectively detect a wide range of previously unseen attacks. The RawNet2 model shows the benefit of end-to-end automatic feature learning, particularly for the *worst-case* A17 attack, with a significant improvement in min t-DCF performance over the baseline.

Chapter 6

In Chapter 6, we present the first application of graph attention networks (GATs) to spoofing and deepfake detection. The work reported in earlier chapters shows that spoofed and bona fide utterances can be distinguished by artefacts in different spectral or temporal domains, which can be detected using models with spectral or temporal attention. However, these approaches often require computationally intensive ensemble systems to detect different forms of attack. The goal of the work presented in Chapter 6 is to design a single, efficient system that can learn the relationship between spoofing cues or artefacts across different spectral and temporal intervals which is capable of detecting a wide range of unseen spoofing attacks without using score-level ensembles. To achieve this, we propose a novel end-to-end spectro-temporal graph attention network, called RawGAT-ST, which uses a GAT to learn the relationships between cues in different sub-bands and temporal intervals. Every node in the graph represents a feature, and each edge indicates the relationship between different node pairs. The proposed method achieved the best reported performance at the time of publication by combining spectral and temporal graphs through addition and multiplication operations.

The work presented in this chapter was published in:

- **Hemlata Tak**, Jee-weon jung, Jose Patino,Massimiliano Todisco, and Nicholas Evans, "**Graph Attention Network for Anti-Spoofing**," in *Proc. INTERSPEECH*, Brno, Czech Republic, September 2021.

- **Hemlata Tak**, Jee-weon jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans, "**End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection**," in *Proc. the ASVspoof 2021 Challenge (INTERSPEECH Satellite Workshop)*, September 2021.

This work introduces the first successful application of graph neural networks and self-attention mechanism to spoofing and deepfake detection.

Chapter 7

Motivated by the impressive performance of the RawGAT-ST model, we explored an extension in the form of a heterogeneous graph attention layer [48]. This leads to a new, integrated spectro-temporal graph attention network named AASIST. The spectral and temporal graphs are heterogeneous, composed of different types of nodes/edges which represent different feature characteristics. The integration of spectral and temporal graph representations using a heterogeneous graph attention layer is shown to be more effective than the approach in the original RawGAT-ST model. At the time of writing, AASIST was the best single CM system reported in the literature.

The work presented in this chapter was published in:

- Jee-weon Jung, Hee-Soo Heo, **Hemlata Tak**, Hye-jin Shim,Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, Nicholas Evans, "**AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks**," in *IEEE ICASSP*, Singapore, May 2022.

This article is the result of joint work with Dr. Jee-Weon Jung (first author). It introduces a more efficient, robust, and integrated spectro-temporal attention network by utilising a novel heterogeneous graph attention layer.

Chapter 8

In Chapter 8, we introduce a novel data augmentation technique, named RawBoost, which operates directly upon raw waveform inputs and which is compatible with the end-to-end models reported in earlier chapters. It helps to reduce model over-fitting and to improve generalisation to more realistic and challenging conditions, such as those in telephony applications.

The work presented in this chapter was published in:

- **Hemlata Tak**, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans, "**RawBoost: A Raw Data Boosting and Augmentation Method applied to Automatic Speaker Verification Anti-Spoofing**," in *Proc. IEEE ICASSP*, Singapore, May 2022.

This work introduces a raw data augmentation technique which uses simple signal processing algorithms to improve generalisation and domain-robustness

in challenging and realistic environments, especially for telephony applications (PSTN+VoIP).

Chapter 9

The final contribution, presented in Chapter 9, relates to our use of self-supervised learning (SSL) through a fine-tuned wav2vec 2.0 pre-trained model to improve model generalisation and domain robustness. We use wav2vec 2.0 as a front-end to extract more generalised and representative features. Despite the SSL front-end being trained initially on a massive amount of only bona fide data, it can substantially improve generalisation to previously unseen spoofing attacks by fine-tuning using in-domain bona fide and spoofed utterances. Results show that the use of an SSL-based front-end in conjunction with RawBoost data augmentation leads to relative improvements of 90% and 88% over the baseline for the LA and DF tasks, respectively, achieving the lowest EERs at the time of writing.

The work presented in this chapter was published in:

- **Hemlata Tak**, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas Evans, "**Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation**," in *Proc. The Speaker and Language Recognition Workshop*, Beijing, China, June 2022.

This work introduce a new state-of-the-art CM solution for spoofing and deepfake detection. The proposed approach leverages self-supervised feature representations with a more sophisticated graph neural network-based classifier, and the RawBoost data augmentation technique.

Publications

Discussed in this manuscript

1. **Hemlata Tak**, Jose Patino, Andreas Nautsch, Nicholas Evans and Massimiliano Todisco, "**An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification**," in *Speaker Odyssey Workshop*, Tokyo, Japan, Nov., 2020.

2. **Hemlata Tak**, Jose Patino, Andreas Nautsch, Nicholas Evans and Massimiliano Todisco, "**Spoofing Attack Detection using the Non-linear Fusion of Sub-band Classifiers**," in *Proc. INTERSPEECH*, Beijing, China, October 2020.

3. **Hemlata Tak**, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans and Anthony Larcher, "**End-to-end anti-spoofing using RawNet2**," in *Proc. IEEE ICASSP*, Toronto, Ontario, Canada, June, 2021.

4. **Hemlata Tak**, Jee-weon jung, Jose Patino,Massimiliano Todisco, and Nicholas Evans, "**Graph Attention Network for Anti-Spoofing**," in *Proc. INTERSPEECH*, Brno, Czech Republic, September 2021.

5. **Hemlata Tak**, Jee-weon jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans, "**End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection**," in *Proc. INTERSPEECH Satellite Workshop of the ASVspoof 2021 Challenge*, September 2021.

6. Jee-weon Jung, Hee-Soo Heo, **Hemlata Tak**, Hye-jin Shim,Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, Nicholas Evans, "**AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks**," in *IEEE ICASSP*, Singapore, May 2022.

7. **Hemlata Tak**, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans, "**RawBoost: A Raw Data Boosting and Augmen-**

tation Method applied to Automatic Speaker Verification Anti-Spoofing," in *Proc. IEEE ICASSP*, Singapore, May 2022.

8. **Hemlata Tak**, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas Evans, "**Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation**," in *Proc. Speaker Odyssey Workshop*, Beijing, China, June 2022.

Other work

1. Jee-weon Jung*, **Hemlata Tak***, Hye-jin Shim, Hee-Soo Heo, Bong-Jin Lee, Soo-Whan Chung, Ha-Jin Yu, Nicholas Evans, Tomi Kinnunen, "**SASV 2022: The First Spoofing-Aware Speaker Verification Challenge**," *Proc. INTERSPEECH*, Incheon, Korea, September 2022.

2. Hye-jin Shim*, **Hemlata Tak***, Xuechen Liu, Hee-Soo Heo, Jee-weon Jung, Joon Son Chung, Soo-Whan Chung, Ha-Jin Yu, Bong-Jin Lee, Massimiliano Todisco, Héctor Delgado, Kong Aik Lee, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, "**Baseline Systems for the First Spoofing-Aware Speaker Verification Challenge: Score and Embedding Fusion**," in *Proc. Speaker Odyssey Workshop*, Beijing, China, June 2022.

   The above two articles relate to the first Spoofing-Aware Speaker Verification Challenge (SASV) Challenge held at INTERSPEECH 2022, of which I was one of the main organisers. My contribution to both articles included preparing baseline scripts, results analysis, and paper writing.

3. Wanying Ge*, **Hemlata Tak***, Massimiliano Todisco and Nicholas Evans, "**On the potential of jointly-optimised solutions to spoofing attack detection and automatic speaker verification**," *Proc. IberSPEECH*, Granada, Spain, November 2022 (Best paper award).

4. Wanying Ge, **Hemlata Tak**, Massimiliano Todisco and Nicholas Evans, "**Can spoofing countermeasure and speaker verification systems be jointly optimised ?**," *Proc. ICASSP*, Rhodes, Greece, June 2023.

   In the above two articles, I was involved in discussing ideas, writing codes, conducting experiments with the lead author and in paper writing.

5. Oubaıda Chouchane, Baptiste Brossier, Jorge Esteban Gamboa Gamboa, Thomas Lardy, **Hemlata Tak**, Orhan Ermis, et. al, "**Privacy-preserving voice anti-spoofing using secure multi-party computation**," in *Proc. INTERSPEECH*, Brno, Czech Republic, September 2021.

I was involved in paper writing and developing a voice anti-spoofing system for the above-mentioned article.

6. Sung Hwan Mun*, Hye-jin Shim*, **Hemlata tak***, Xin Wang, Md Sahidullah, Min Hyun Han, Myeonghun Jeong, Xuechen Liu, Massimiliano Todisco, Kong Aik Lee, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, Nam Soo Kim and Jee-weon Jung, "**Towards single integrated spoofing-aware speaker verification embeddings**," accepted in INTERSPEECH, Dublin, Ireland, August 2023.

7. Michele Panariello, Wanying Ge, **Hemlata Tak**, Massimiliano Todisco and Nicholas Evans, "**Malafide: A novel adversarial convolutive noise attack against deepfake and spoofing detection systems**," accepted in INTERSPEECH, Dublin, Ireland, August 2023.

For the two article above, I was involved in discussion and paper writing.

Journal

1. Tomi Kinnunen, Kong Aik Lee, **Hemlata Tak**, Nicholas Evans and Andreas Nautsch, "t-EER: Parameter-Free Tandem Evaluation of Countermeasures and Biometric Comparators," *under revision in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

   In the aforementioned journal article, my contribution was conducting experiments and preparation of results.

Chapter 2

Literature review

Spoofing attacks pose a significant threat as they allow fraudsters to gain unauthorised access to resources, services or devices. No matter what the security level, for the most part, these threats can be unacceptable. To address this problem, the ASVspoof initiative has led efforts to design spoofing countermeasures (CMs), which aim to automatically detect and deflect spoofing attacks against ASV system. This effort was initiated following the first special session on anti-spoofing [14] held at INTERSPEECH in 2013.[1] The ASVspoof initiative has since, collected and distributed large scale databases comprising both genuine and spoofed utterances generated using a range of advanced algorithms. These databases have been used in biennial challenges [15, 40, 41, 49], which have yielded significant improvements in the reliability of spoofing detection.

This chapter provides a description of the ASVspoof corpus and evaluation metrics, as well as a brief overview of the background literature relevant to the research work presented in this thesis. It includes a detailed review of CMs used for spoofing detection for the logical access (LA) and deepfake (DF) detection tasks. For a more detailed summary on automatic speaker verification spoofing detection, readers can refer to widely cited survey papers [8, 29, 38, 39].

2.1 Spoofing databases

Presented in this section are the databases used for the design and evaluation of CMs against spoofing attacks. The first ASVspoof challenge in 2015 [15] studied a LA scenario. For the 2017 challenge [40], the goal was to detect replayed speech in a physical access (PA) scenario. In 2019 [49, 50], more advanced, state-of-the-art text-to-speech synthesis (TTS) and voice conversion (VC) algorithms were used for a LA spoofing detection task. The most recent 2021 edition [41, 51], involved the

[1] https://www.asvspoof.org/

most challenging scenario to date, involving variable transmission and encoding conditions as well as a new DF detection task in which there is no ASV. This thesis reports experiments with three of these databases, namely the ASVspoof 2015, 2019 and 2021 LA and DF databases. They were all generated from the same VCTK source database.[2]

2.1.1 ASVspoof 2015 logical access database

The ASVspoof 2015 database[3] was the first challenge database, released in 2015, for automatic speaker verification spoofing detection. It contains three disjoint partitions: training, development and evaluation. Each partition comprises a set of genuine (bona fide) and spoofed utterances, with the latter generated with TTS and VC algorithms [52] which were popular at that time. There are a total of 10 spoofing attacks. The first 5 attacks (S1–S5) are used in generating the training and development partition and are collectively referred to as *known* attacks. The remaining attacks (S6–S10) are used in generating the evaluation partition and are referred to as *unknown* attacks. Table 2.1 summarises the statistics of each subset. Full details of the database and protocols are available in [52].

Table 2.1: Statistics of the database used in ASVspoof 2015 challenge.

Subsets	# Speakers		# Utterances	
	Male	Female	Bona fide	Spoof
train	10	15	3750	12625
development	15	20	3497	49875
evaluation	20	26	9404	184000

2.1.2 ASVspoof 2019 logical access database

Other experiments in this thesis were conducted using the ASVspoof 2019 challenge database[4] which consist of three disjoint partitions: training, development and evaluation. Spoofed utterances in each partition are generated using a set of more advanced VC, TTS, and hybrid (VC-TTS) algorithms [53]. There are a total of 19 different spoofing attacks. Attacks in the training and development partition were created with a set of 6 different algorithms (A01-A06), whereas attacks in the evaluation partition were created with a set of 13 algorithms (A07-A19). Four TTS algorithms (A01-A04) and two VC algorithms (A05 and A06) were used to generate spoofing attacks in the training and development partitions. Two TTS

[2] http://dx.doi.org/10.7488/ds/1994
[3] https://datashare.ed.ac.uk/handle/10283/782
[4] https://doi.org/10.7488/ds/2555

Table 2.2: A summary of the spoofing attack algorithms (VC and TTS based) from the ASVspoof 2019 logical access database. * indicates neural networks.

			Attack type	**Conversion**	**Waveform generation**
Attacks in train and Dev.		A01	TTS	AR RNN*	WaveNet *
		A02	TTS	AR RNN*	WORLD
		A03	TTS	FF*	WORLD
		A04	TTS	CART	Waveform concat.
		A05	VC	VAE*	WORLD
		A06	VC	GMM-UBM	Spectral filtering + OLA
Attacks in Eval	Known attacks	A16	TTS	CART	Waveform concat.
		A19	VC	GMM-UBM	Spectral filtering + OLA
	Partially known attacks	A07	TTS	RNN*	WORLD
		A08	TTS	AR RNN*	Neural source-filter*
		A09	TTS	RNN	Vocaine
		A17	VC	VAE*	Waveform filtering
	Unknown attacks	A10	TTS	AR RNN+ CNN	WaveRNN*
		A11	TTS	AR RNN+ CNN	Griffin-Lim
		A12	TTS	RNN	WaveNet *
		A13	VC-TTS	Momentum match*	Waveform filtering
		A14	VC-TTS	RNN*	STRAIGHT
		A15	VC-TTS	RNN*	WaveNet *
		A18	VC	Linear	MFCC vocoder

algorithms (A07 and A16), four VC algorithms (A13, A14, A17, and A19), and two hybrid algorithms (VC with TTS-generated inputs, A13 and A14) were used to generate the evaluation partition. Table 2.2 summarises the waveform conversion and generation techniques used for each spoofing attack. More detailed information on each attack algorithm can be found in [53]. It is important to note that in the 2019 LA evaluation set, the spoofing attacks A16 and A19 can be considered as *known* attacks because the same algorithms were used to generate spoofed trials for the training and development partitions (A04 and A06), despite the data corresponding to disjoint utterances and speakers. The remaining spoofing attacks in the test set are either completely *unknown* or only partially similar to those used in the training set. The statistics of the ASVspoof 2019 database are summarised in Table 2.3.

2.1.3 ASVspoof 2021 logical access database

As per the evaluation plan [41, 51], ASVspoof 2019 LA training and development data were utilised for the training and optimisation of CM solutions for

Table 2.3: Statistics of the database used in ASVspoof 2019 and ASVspoof 2021 challenge. ASVspoof 2019 LA training and development partitions were used in ASVspoof 2021 LA and DF challenge.

Subset	Speaker		ASVspoof 2019 LA		ASVspoof 2021 LA		ASVspoof 2021 DF	
	Male	Female	Bona fide	Spoof	Bona fide	Spoof	Bona fide	Spoof
Train	8	12	2580	22800	-	-	-	-
Dev	4	6	2548	22296	-	-	-	-
Eval	21	27	7355	63882	14816	133360	14869	519059

the ASVspoof 2021 challenge. The use of any other external speech data was forbidden. The ASVspoof 2021 LA evaluation database[5] was sourced from the ASVspoof 2019 evaluation partition. While both ASVspoof 2019, and hence also the 2021 LA databases are generated from the same VCTK source database [44], some utterances in the 2021 evaluation database were transmitted across a telephony network such as a PSTN or VoIP, using various codecs including A-law, G.722, and others [41]. This resulted in the seven LA evaluation conditions illustrated in Table 2.4. The codecs operating at 8 kHz includes C2:a-law, C4: μ-law, and C5: GSM. Conditions C1:none, C4: G.722 and C7:OPUS correspond to a 16 kHz sampling rate. The bit rates of each codec are listed in the last column of Table 2.4. ASVspoof 2021 LA data is divided into two subsets: progress and evaluation. The progress subset contains a small number of trials from known codec conditions (C1-C4) and was used for intermediate assessment before the final evaluation submission. The evaluation subset contains the remaining trials from C1-C4 conditions in addition to all trials from unknown conditions (C5-C7). The statistics of ASVspoof 2021 LA and DF database are summarised in Table 2.3.

Table 2.4: Summary of LA evaluation conditions. Codecs in *Italics* are unknown only used in the evaluation phase.

Cond.	**Codec**	**Sampling rate**	**Bitrate**
LA-C1	no codec	16 kHz	250 kbps
LA-C2	a-law	8 kHz	64 kbps
LA-C3	μ-law	8 kHz	64 kbps
LA-C4	G.722	16 kHz	64 kbps
LA-C5	μ-law	8 kHz	64 kbps
LA-C6	GSM	8 kHz	13 kbps
LA-C7	OPUS	16 kHz	VBR 16 kbps

[5] https://doi.org/10.5281/zenodo.4837263

Table 2.5: Summary of DF evaluation conditions. Italics only codecs used in the evaluation phase.

Cond.	Compression	Bitrate
DF-C1	-	256 kbps
DF-C2	low mp3	∼80-120 kbps
DF-C3	high mp3	∼220-260 kbps
DF-C4	low m4a	∼20-32 kbps
DF-C5	high m4a	∼96-112 kbps
DF-C6	low ogg	∼80-96 kbps
DF-C7	high ogg	∼256-320 kbps
DF-C8	mp3→ m4a	∼80-120 kbps, ∼96-112 kbps
DF-C9	ogg→ m4a	∼80-96 kbps, ∼96-112 kbps

2.1.4 ASVspoof 2021 deepfake database

The ASVspoof 2021 DF database[6] [41] is based upon multiple diverse corpora, including the VCTK database [44], the 2018 [45] and 2020 [46] voice conversion challenge (VCC) databases, and includes spoofed utterances generated with over 100 diverse spoofing attack algorithms. The aim of the DF task is to address CM generalisation across different compression algorithms as well as different domains (different databases) and unknown spoofing attacks. Audio utterances were processed with the set of different media codecs, giving the DF evaluation conditions listed in Table 2.5: DF-C1: no codecs, DF-C2 and C3: mp3 codec, DF-C4, and C5: m4a codec, DF-C6 and C7: ogg vorbis codec with different, variable bit rates, and DF-C8: mp3→m4a, DF-C9: ogg→m4a. The last two dual codec conditions were included to determine whether spoofing artefacts can still be detected after transcoding. Each of these codec conditions include different vocoder conditions (from the 2018 [45] and 2020 [46] VCC databases) such as traditional, neural autoregressive, neural non-autoregressive, and waveform concatenation [54]. The DF database is more challenging in terms of generalisation as audio utterances are from unknown domains and are generated with far more diverse spoofing attacks. The ASVspoof 2019 training partition contains neither encoding, transmission nor media compression effects. There is hence an interest in data augmentation techniques to compensate for the lack of in-domain training and development data [41]. Full details of both LA and DF database and experimental protocols are available in [41, 54].

[6] https://doi.org/10.5281/zenodo.4835108

2.2 Performance metrics

Spoofing detection is usually approached as a binary classification task in which the classifier evaluates the input trial and assigns a score to each trial. If the score is greater than some pre-defined threshold τ_{cm}, then the trial is classified as bona fide, otherwise spoofed. The most common metric used for evaluation is the Equal Error Rate (EER). The EER is defined by an operating point at which both the false acceptance rate (P_{fa}^{cm}, spoofed trials classified as bona fide) and the false rejection rate (P_{miss}^{cm}, bona fide trials classified as spoofed) are equal. For all experiments reported in this thesis, the EER is computed using a convex hull approach with the Bosaris toolkit [55]. The P_{fa}^{cm} and the P_{miss}^{cm} are defined as follows:

$$\mathrm{P}_{\mathrm{fa}}^{\mathrm{cm}}(\tau_{\mathrm{cm}}) = \frac{\#\text{spoofed trials with CM scores} > \tau_{\mathrm{cm}}}{\#\text{total spoofed trials}} \tag{2.1}$$

$$\mathrm{P}_{\mathrm{miss}}^{\mathrm{cm}}(\tau_{\mathrm{cm}}) = \frac{\#\text{bona fide trials with CM scores} \leq \tau_{\mathrm{cm}}}{\#\text{total bona fide trials}} \tag{2.2}$$

False accepts (FAs) occur when a spoofed trial is assigned a score greater than τ_{cm} and is hence accepted, whereas false rejects (FRs) occur when a bona fide trial is assigned a score less than or equal to τ_{cm} and is rejected. Lower EER values are an indication of better discrimination performance. Although the EER has been deprecated in recent ISO/IEC standards [5, 6], it is still used as a convenient evaluation metric within the speaker recognition community.

The EER metric is used to assess standalone CM performance and, being a parameter-free (no priors or detection costs), it is be unrepresentative of performance in practical applications. To gauge the impact of spoofing and CMs upon the reliability of an ASV system, a new metric was introduced in 2018. The minimum tandem detection cost function (min t-DCF) [56, 57] is an ASV-constrained evaluation metric for the assessment of ASV and CMs operating in tandem and was used as the primary evaluation metric in both the 2019 and 2021 ASVspoof challenges. The t-DCF metric is an extension of the detection cost function (DCF) that is widely used for the evaluation of ASV performance [58]. Both the ASV system and the detection threshold τ_{asv} are fixed by the ASVspoof organisers, while the CM threshold is variable according to the CM system developed by participants. The min t-DCF is defined as:

$$\text{min t-DCF} = \min_{\tau} \frac{C_0 + C_1 P_{\mathrm{miss}}^{\mathrm{cm}}(\tau_{\mathrm{cm}}) + C_2 P_{fa}^{\mathrm{cm}}(\tau_{\mathrm{cm}})}{C_0 + \min(\{C_1, C_2\})} \tag{2.3}$$

where coefficients C_0, C_1, and C_2 are hyper-parameters computed from ASV scores, the prior probabilities of target, non-target, and spoofed trials, and ASV and CM

detection costs. The coefficients are defined as follows:

$$\begin{aligned} C_0 &= \pi_{\text{tar}} C_{\text{miss}}^{\text{asv}} P_{\text{miss}}^{\text{asv}} + \pi_{\text{non}} C_{\text{fa}}^{\text{asv}} P_{\text{fa}}^{\text{asv}} \\ C_1 &= \pi_{\text{tar}} C_{\text{miss}}^{cm} - (\pi_{\text{tar}} C_{\text{miss}} P_{\text{miss}}^{\text{asv}} + \pi_{\text{non}} C_{\text{fa}}^{\text{asv}} P_{\text{fa}}^{\text{asv}}) \\ C_2 &= \pi_{\text{spoof}} C_{fa,spoof}^{cm} P_{\text{fa,spoof}}^{\text{asv}} \end{aligned} \tag{2.4}$$

where: $C_{\text{miss}}^{\text{asv}}$ is the cost of the ASV system rejecting a target trial; $C_{\text{fa}}^{\text{asv}}$ is the cost of the ASV system accepting a nontarget trial; $C_{\text{miss}}^{\text{cm}}$ is the cost of the CM rejecting a bona fide trial; $C_{\text{fa}}^{\text{cm}}$ is the cost of the CM accepting a spoofed trial. π_{tar}, π_{non} and π_{spoof} are prior probabilities of target, non-target and spoofed trials. The denominator in Eq. 2.3 normalises the min t-DCF so that its value ranges between 0 and 1. A t-DCF value of 0 implies perfect (error-free) ASV and CM systems, whereas a value of 1 or higher indicates that the CM system offers imperfect or no protection against spoofing attacks. The lower bound is referred to as the ASV floor ($\frac{C_0}{C_0+\min(C_1,C_2)}$) and is the value of the t-DCF for an imperfect ASV system (provided by the challenge organisers) but a perfect CM. Similar to the EER, lower t-DCF values indicate better performance. The reader is referred to [56, 57] for further details.

2.3 Spoofing detection

The goal of spoofing detection is to distinguish between genuine (bona fide) and spoofed speech. It traditionally comprises two stages: a front-end feature extractor and a back-end classifier. Continuous advances in deep learning and machine learning have enabled the development of new and advanced algorithms for generating synthetic speech, making it difficult to detect spoofed speech. To overcome these difficulties, a variety of detection systems have been proposed in literature to prevent ASV systems from being manipulated. This section presents an overview of the vulnerabilities of ASV to spoofing attacks and the CMs that have been explored in the past three editions of the ASVspoof LA challenge (i.e., ASVspoof 2015, 2019 and 2021) as well as in post challenge studies. More comprehensive reviews of recent advances in LA spoofing detection using the same benchmark datasets can be found in [59, 60].

2.3.1 ASVspoof 2015 logical access task

In order to verify vulnerabilities to spoofing, ASVspoof organisers evaluated an ASV baseline system using ASVspoof 2015 evaluation data. Results are illustrated in Table 2.6, reproduced from [15]. The ASV baseline utilised an i-vector [61] embedding extractor with probabilistic linear discriminant analysis (PLDA) [62] as a back-end. The first row indicates ASV baseline performance. Rows 2-11 present performance for the same ASV system when subjected to

Table 2.6: ASV baseline performance when subjected to various spoofing attacks for ASVspoof 2015 database..

Attack	**EER** (%)
PLDA (ASV)	2.30
S1	32.55
S2	2.66
S3	40.29
S4	43.35
S5	46.24
S6	44.71
S7	29.29
S8	36.19
S9	33.53
S10	**51.17**
Avg (S1-S10)	36.00

different spoofing attacks. Performance degrades significantly (2.30% to 51.17% EER for the S10 attack). These results confirm vulnerabilities to spoofing and demonstrate the need for robust CMs.

The ASVspoof 2015 challenge attracted the submission of 16 primary (fusion-based) submissions. While most CM submissions achieved an EER below 1% for known attacks, none generalised well to unknown attacks. The top-performing system, as described in [63], employed two different front-end features: Mel-Frequency Cepstral Coefficients (MFCC) and Cochlear Filter Cepstral Coefficients with change in Instantaneous Frequency (CFCC-IF). This system achieved an average EER of 0.41% for known attacks, and 2.01% for unknown attacks, for a combined average EER of 1.21%. This trend was consistent among the top primary submissions, which showed higher EERs for unknown attacks than for known attacks. One explanation for this disparity is the difficulty in detecting the unknown S10 attack, which was implemented using a unit selection-based waveform concatenation TTS algorithm. Table 2.7 provides a summary of results for top-performing solutions including ASVspoof 2015 challenge submissions and as well as post-challenge results.

As illustrated in Table 2.7, all top-performing CM systems use the fusion of dif-

Table 2.7: Comparison of CM systems for the ASVspoof 2015 LA task. There were some top-performing challenge submissions and other results are from post-challenge studies. Results reported in terms of average EER (%) of all known and unknown attacks.

Challenge results					
Ref.	front-end	back-end	EER (%)		
			Known	Unknown	Avg.
[63]	MFCC, CFCC-IF	GMM	0.40	2.01	1.21
[64]	MFCC, MFPC,CosPhase	SVM	0.01	3.92	1.97
[65]	F-bank energies	DNN	0.06	4.99	2.52
[66]	LMS,RLMS,MGD,IF	MLP	0.01	5.23	2.61
[67]	MFCC+MGDFCC+WLP-GDCC	GMM	0.04	5.34	2.69
Post-challenge results					
Ref.	front-end	back-end	EER (%)		
			Known	Unknown	Avg.
[68]	raw-waveform	MLP	0.03	5.75	2.89
[69]	GD	RNN	0.40	3.33	1.86
[70]	Spec	CNN	0.16	2.64	1.40
[16]	RFCC	GMM	0.12	1.92	1.02
[16]	LFCC	GMM	0.11	1.67	0.89
[71]	CQCC	GMM	0.05	0.46	0.25

ferent magnitude and phase-based features. These include Mel-frequency cepstral coefficients (MFCC) [72], Rectangular frequency cepstral coefficients (RFCC) [16], CFCC-IF [63], group delay (GD), modified group delay (MGD) [73], relative-phase, and instantaneous frequency derivatives (IF) [73]. Additionally, other front-ends, such as spectrogram (Spec), Mel-filter bank (FBank), and linear filter bank (LFB) outputs, were also explored. All top-performing systems demonstrate excellent results in the detection of known attacks as shown in Table 2.7, with EERs consistently below 0.5%. However, EERs for unknown attacks are notably higher and all above 2%. The performance gap between known and unknown attacks is significant and highlights the challenge to develop generalised CMs. In order to improve generalisation, a novel front-end based on constant Q cepstral coefficients (CQCCs) [71,74] was proposed in 2016 (post-challenge) and is now widely used as a default front-end for spoofing detection. CQCCs are derived using the constant Q transform (CQT), and improve substantially upon the performance of ASVspoof 2015 challenge submissions. A CQCC front-end with a Gaussian mixture model (GMM) classifier obtained the lowest average EER of 0.25% for the ASVspoof 2015 challenge database at the time of publication. CQCC features with a GMM classifier was adopted as a baseline in subsequent ASVspoof editions.

Table 2.8: ASV baseline performance when subjected to various spoofing attacks for ASVspoof 2019 LA database.

Attack system	Development	Evaluation
ASV (x-vector)	2.43	2.48
A01	24.52	-
A02	15.04	-
A03	56.94	-
A04	**63.02**	-
A05	21.90	-
A06	10.11	-
A07	-	59.68
A08	-	40.39
A09	-	8.38
A10	-	57.73
A11	-	59.64
A12	-	46.18
A13	-	46.78
A14	-	64.01
A15	-	58.85
A16	-	**64.52**
A17	-	3.92
A18	-	7.35
A19	-	14.58

2.3.2 ASVspoof 2019 logical access task

The results of the ASVspoof 2015 challenge [15] show that CM designs primarily focused on identifying salient features rather than investigating advanced neural network-based classifiers. The ASVspoof 2019 challenge and subsequent studies shifted towards more complex and better performing neural network-based classifiers. The minimum tandem Decision Cost Function (min t-DCF) [56] (described in Section 2.2) was introduced as the default evaluation metric. The ASVspoof 2019 challenge showed the detrimental effects of advanced TTS and VC-based spoofing attacks on ASV performance, as shown in Table 2.8 (results reproduced from [53]). The ASV baseline utilised an x-vector [75] embedding extractor with PLDA [62] as a back-end. The first row in Table 2.8 illustrates ASV baseline performance with target and non-target (impostor) bona fide trials, while the remaining rows depict results with target and spoofing attack trials. The severity of the spoofing attacks varies among the different attack algorithms. For e.g. the TTS-based (A04) attack generated with a waveform concatenation method increases the ASV EER from 2.43% (target *vs.* non-target) to 63.02% (target *vs.* A04 attack) on the development set. Attack A16 increases the ASV EER from 2.48% to 64.52% on the evaluation set. There attacks were found to

Table 2.9: Empirical description of top-5 submissions for ASVspoof 2019 LA task.

Team ID	Front-end features	Classifiers	System Fusion
T05	DFT, DCT,	MobileNet, DenseNet ResNet50	Weighted score average
T45	LFCC, CQT, DFT	GMM-UBM, LCNN	Weighted score average
T60	MFCC,IMFCC,SCMC Log-DFT, Mel-spec raw audio, CQCC	GMM-UBM,CNN CRNN,Wave-U-Net SVM	Logistic regression
T24	CQCC, LFB	Resnet18, NN layer	Score fusion
T50	Log CQT, Phase-gram	VAE, CGCNN CGRNN, ResNet18	Score averaging

be the most damaging to ASV performance.

The ASVspoof 2019 challenge LA challenge attracted 48 submissions out of which 27 submissions outperformed the strong B02 baseline (LFCC-GMM) [49]. Table 2.9 provides a summary of the top-5 challenge submissions, which are mostly ensembles of several complex deep neural network (DNN) based sub-systems. The top-performing T05 system used a fusion of seven sub-models, including four MobileNets [76], one DenseNet [77, 78], and two ResNet50 [79] networks. Other teams explored more efficient DNN architectures such as a light convolutional neural network (LCNN) [80], variational auto-encoders (VAEs), complex gated convolutional networks (CGCNN) [81], and Wave-U-Net [82] models. Among these, LCNNs have proven to be particularly efficient and are widely used for spoofing and deepfake detection. LCNN models use a max-feature-map (MFM) non-linearity function to select optimal feature maps during training and achieved the best single system (T45) performance in the ASVspoof 2019 challenge for the LA task [83]. Full details related to the ASVspoof 2019 challenge results and findings can be found in [47, 49].

Among the post-challenge studies, front-end approaches that use CQCCs, CQT, LFCCs and learnable audio front-end features [84] with neural network-based back-ends, such as CNN, LCNN, ResNet, CapsuleNet and LSTM, have shown improved performance [83, 85–91]. Other notable work includes one-class learning approaches [92–94] which model the distribution of genuine (bona fide) utterances only. They can still be used as two-class classifiers and to reject utterances that do not produce a sufficiently high likelihood when compared to the model. In

this case, spoofing detection becomes an out-of-distribution detection or anomaly detection problem. A new loss function, called the one-class (OC) softmax function [93], was introduced in 2021. It compacts the bona fide utterance representation and injects an angular margin to separate bona fide utterances from spoofing attacks in the latent space. One-class learning approaches also perform reliably for spoofing and deepfake detection [93, 94].

Another interesting approach to improve CM generalisation and robustness to different channel variability for in-the-wild scenarios is reported in [95]. It shows a significant degradation in CM performance when a model trained on the ASVspoof 2019 LA database is evaluated using out-of-domain databases, such as the ASVspoof 2015 and 2020 voice conversion challenge (VCC) evaluation databases. The CM EER increased from 2.09% for ASVspoof 2019 evaluation set to 26.03% for ASVspoof 2015 evaluation set, and to 41.66% for 2020 VCC dataset. The substantial degradation in CM performance is likely due to the channel-mismatch between training and testing conditions. This work shows the importance of developing CMs which are robust to channel variability as well as different spoofing attacks. This scenario was explored in the most recent ASVspoof 2021 LA challenge [41].

2.3.3 ASVspoof 2021 logical access task

Both the 2015 and 2019 ASVspoof LA databases contain clean utterances without any background noise or channel variation. Such ideal conditions are not realistic. Several studies have shown the significant degradation in CM performance when they are deployed in more realistic conditions, e.g. involving telephony transmission [42, 43]. To address this problem, the LA task of the ASVspoof 2021 challenge [41] was designed to foster improvements in spoofing detection reliability in the face of nuisance variation stemming from unknown encoding and telephony transmission conditions. The LA task involved more realistic conditions in which audio data is passed through telephony or VoIP networks. The codecs used in generating the ASVspoof 2021 LA database are a combination of traditional codecs, such as A-law, μ-law and G.722, and more modern codecs like OPUS. The default metric remained the min t-DCF [56], with the EER serving as a secondary metric.

ASVspoof 2021 challenge participants applied various techniques such as different transmission codecs, compression algorithms, and low-pass filtering in the form of data augmentation (DA) to the original ASVspoof 2019 LA training data [87, 95, 98, 101, 103–106]. Table 2.10 summarises the top-performing submissions for LA and DF tasks. Results are presented separately for two subsets: progress and

Table 2.10: A summary of results for top-performing systems for ASVspoof 2021 LA and DF tasks. Results reported in terms of pooled EER (%). DA: data augmentation techniques.

Ref.	**Feat.**	**Model**	**DA**	**LA**		**DF**	
				progress	**eval**	**progress**	**eval**
[96]	MSTFT, Raw	LCNN ResNet, LSTM	mixup codec, FIR	0.89	1.32	0.24	15.64
[97]	LFB	ResNet, MLP	FM, codec, MUSAN	2.39	3.21	1.79	16.05
[98]	CQT	LCNN	codec	3.23	3.62	2.93	18.30
[99]	LFCC	ECAPA	codec	5.13	5.46	6.88	20.33
[100]	LFCC	ResNet18	codec	6.21	6.62	0.78	23.13
[101]	Raw	RawNet2	codec	5.79	6.36	-	-
[102]	Raw	RawNet2	codec	7.88	8.05	-	-

evaluation. The top-performing system [96] explored DA using FIR filtering to emulate the application of different telephony codecs. This approach obtained the lowest EER of 1.32% for the ASVspoof 2021 LA evaluation set. More traditional DA techniques, such as SpecAugment [107], the introduction of additive noise using the MUSAN database [108], the introduction of convolutive noise using the room impulse response (RIR) database [109], and telephony codec augmentation [98], were more popular. All these DA techniques utilise additional data resources. As illustrated in Table 2.10, the top-performing challenge entries used some form of spectral or cepstral features, and raw waveform inputs and efficient LCNN, ResNet and TDNN back-end classifiers. Most ensemble systems explored traditional fusion approaches, including weighted score averaging and logistic regression.

2.3.4 ASVspoof 2021 deepfake task

The ASVspoof 2021 challenge introduced a new deepfake (DF) detection task, in addition to LA and PA tasks. It focuses on spoofing and deepfake detection in the presence of general audio compression. The DF database consists of audio utterances collected from different sources which are compressed with different lossy media codecs, e.g. mp3 and m4a. Utterances are first compressed using a codec and then decoded to recover uncompressed audio. The compression and decoding is lossy and introduces distortions that vary based on the type of codec and its configuration, distortions which impact both bona fide and spoofed utterances and which can mask spoofing artefacts. This scenario mimics a detection task when deepfakes are posted to social media, or broadcast on television. Unlike for the LA task, for the DF task there is no ASV system. Accordingly, the CM EER is used as the default performance metric. As shown in Table 2.10, the top-performing system obtained an EER of 0.24% on the progress set and 15% EER for the evaluation subset. This performance gap indicates that models trained solely on ASVspoof 2019 LA training data do not generalise well to previously unseen attacks and out-of-domain data, resulting in degraded performance. This is a consequence of over-fitting. For more detailed information on the ASVspoof 2021 LA and DF challenge results and findings, readers are referred to [54].

Table 2.11 presents a summary of the recent trends observed in all three editions of the ASVspoof challenge which focused on LA attacks. The first edition, ASVspoof 2015 challenge generated spoofed utterances using unit-selection TTS and vocoders. The 2019 edition built on this with more advanced neural and acoustic waveform models, such as Tacotron2 [23], WaveNet [19,110], and waveform concatenation algorithms. The recent ASVspoof 2021 LA challenge [41] incorporated both the 2015 and 2019 editions with more realistic telephony and compression

Table 2.11: Summary of the trends in three ASVspoof challenges which tackled the logical access (LA) conditions.

	ASVspoof 2015	**ASVspoof 2019 LA**	**ASVspoof 2021 LA**
Attacks	TTS (diphone concat.), VC (STRAIGHT vocoder)	TTS (Tacotron2,Waveform concat.), VC (Waveform filtering)	Waveform concat.,Tacotron2, neural vocoders
Pre-processing	Pre-emphasis	-	Trans codec, mp3 compression, speed perturbation, mixup, RIR and MUSAN
Front-ends	MFCC, MGD, LPC, PLP, CFCC-IF, IMFCC, i-vector	CQCC, LFCC, STFT, CQT, LFB, Raw waveform	LFCC, SSL, Mel Spec, LFB, Raw waveform
Back-ends	GMM, MLP, SVM, RNN, PLDA, GMM-UBM	CNN, CapsuleNet, ResNet, LSTM, LCNN, MobileNet	LCNN, ResNet, transformer VAE,SENet, ECAPA-TDNN
Post-processing	CMVN feature norm., score-average	weighted score average	weighted score average

conditions. The trends in front-end features and back-end classifiers also shifted towards more advance front-end features and neural network-based classifiers, such as from GMM, DNN, MLP in the 2015 edition to LCNN, ResNet, Siamese and capsule neural networks in the 2019 edition. These advances were further developed in the recent 2021 challenge, which includes emphasized channel attention propagation and aggregation time delay neural networks (ECAPA-TDNN) [111], and ResNet with the squeeze-and-excition (SENet) [112] neural networks.

2.4 Summary

This chapter provides an overview of the ASVspoof challenge series, and highlights the advances and goals specific to each edition. We also described the tandem detection cost function and the databases used for experimental work presented later in this thesis. Finally, we provide a brief overview of the recent trends in CM development across all ASVspoof editions with a focus on LA scenarios.

Chapter 3

An Explainability Study of the CQCC Front-end

One of the best performing front-end features, known as Constant Q Cepstral Coefficients (CQCCs) [71] proposed in 2016, have been shown to be especially effective in detecting spoofing attacks implemented with a unit selection-based speech synthesis algorithm [20]. Despite their success, they largely fail in detecting other forms of spoofing attacks, where more traditional Linear Frequency Cepstral Coefficient (LFCC) [16] front-end representations give substantially better performance. We also observed similar differences in the ASVspoof 2019 challenge submissions and baseline results [49]. These observations have led us to ask ourselves why? What could account for these performance differences? The work presented in this chapter aims to revisit CQCC front-end features and attempts to explain why they are effective in detecting some attacks but less effective in detecting others. This chapter aims to shed light upon what is being detected in the signal, i.e. the signal or feature-level artefacts that serve to distinguish different forms of spoofing attack from bone fide speech. The explanation at this level would surely help us to go beyond CQCC feature and to design more reliable countermeasures (CMs) that can detect a broader range of attacks.

This work aims to investigate the effectiveness of a CQCC front-end in detecting different forms of spoofing attacks for speaker verification. To this end, we perform a sub-band analysis to understand where information relevant to spoofing detection is located in the spectrum, and we evaluate the performance of CQCCs on the ASVspoof 2015 and 2019 LA databases. We also examine the impact of a variable spectro-temporal resolution on spoofing detection performance.

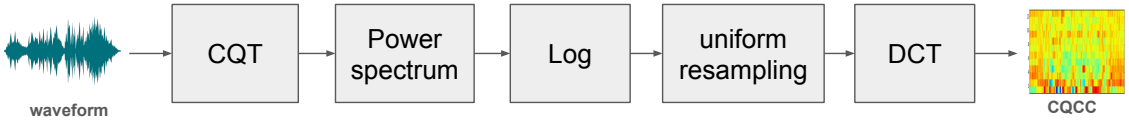


Figure 3.1: Block diagram of CQCC feature extraction.

3.1 Constant Q cepstral coefficients

In this section, we explain the rationale for utilising CQCCs and the process of deriving them from the constant Q transform (CQT). Additionally, we present a comparison of CQCCs to discrete Fourier transform (DFT) derived features and their performance on the ASVspoof 2015 database.

3.1.1 Motivation

The fundamental motivation behind the development of CQCCs was that, since attackers try to replicate the same features used for ASV when generating spoofed speech, features not optimised for ASV should have a better potential for spoof detection. CQCCs are derived using the CQT, which is commonly used in music processing and reflects human perception more closely than the linear scale used in the discrete Fourier transform [113]. Although CQCCs have also been found to be useful for other speech tasks including speaker diarization, ASV and utterance verification [74, 114], they were not specifically designed for ASV.

3.1.2 From the CQT to CQCCs

The perceptually motivated CQT [115, 116] approach to the spectro-temporal analysis of a discrete signal $x(n)$ is defined by:

$$X_{CQ}(k,n) = \sum_{l=n-\lfloor N_k/2 \rfloor}^{n+\lfloor N_k/2 \rfloor} x(l) s_k^*(l - n + N_k/2) \tag{3.1}$$

where n is the sample index, $k = 1, 2, ..., K$ is the frequency bin index, $s_k(n)$ are the basis functions, $*$ is the complex conjugate and N_k is the frame length. The basis functions $s_k(n)$ are defined by:

$$s_k(n) = g_k(n) e^{j2\pi n f_k / f_s}, n \in \mathbb{Z} \tag{3.2}$$

where $g_k(n)$ is zero-centred window function, f_k is the center of each frequency bin and f_s is the sampling rate. Further details of the CQT algorithm are available in [117].

The centre of each frequency bin f_k is defined according to $f_k = 2^{(k-1)/(B)} f_1$, where f_1 is the centre of the lowest frequency bin and B is the number of bins per octave. B determines the trade-off between spectral and temporal resolutions. The value of $N_k \in \mathbb{R}$ in Eqs. 3.1 and 3.2 is a real number and inversely proportional to f_k. As a result, the summation in Eq. 3.1 is over a number of samples that is dependent upon the frequency. Hence, the spectro-temporal resolution is also dependent on the frequency. The quality (Q) factor, given by $Q = f_k/\delta f$, where δf is the bandwidth, reflects the selectivity of each filter in the filter bank. For the CQT transform, Q is constant for all frequency bins k; filters are logarithmically spaced. The CQT gives $X_{CQ}(k, n)$ a geometrically spaced spectrum, whereas the basis functions of the DCT are linearly spaced. As a result, the geometric DCT basis is no longer orthogonal. To address this, the non-uniform frequency scale of the CQT is resampled using a spline interpolation method to a uniform, linear scale, giving $\bar{X}_{CQ}(l, n)$ which attributes equal weighting to information across the full spectrum [71]. CQCCs are obtained from the discrete cosine transformation (DCT) of the logarithm of the squared-magnitude CQT as follows:

$$CQCC(p, n) = \sum_{l=0}^{L-1} \log \left| \bar{X}_{CQ}(l, n) \right|^2 \cos \left[\frac{p \left(l - \frac{1}{2} \right) \pi}{L} \right] \tag{3.3}$$

where $p = 0, 1, ..., L - 1$, and where l is now the linear-scale frequency bin index. The feature extraction process of CQCCs is summarised in the Figure 3.1. Full details of the CQCCs extraction algorithm can be found in [71]. Efficient implementations of the CQT can be found in [117] and [118].

3.1.3 Differences between DFT and CQT

The main differences between DFT and CQT relate to their spectro-temporal resolution. Essentially, spectral decomposition acts as a filter bank, but the CQT and DFT have different properties. In contrast to the CQT, the set of filter bank frequencies in the DFT are linearly distributed and the bandwidth of each filter is constant. Additionally, the Q factor in the DFT is no longer constant; it increases linearly as the frequency increases. The series of filters in the DFT is no longer logarithmically spaced, but linearly spaced, which ensures that the DFT exhibits constant spectral resolution. This is not the case for the CQT. Compared to the DFT (linear frequency scale), the CQT-derived spectrum has greater frequency resolution at lower frequencies than at higher frequencies, as a direct consequence of the constant Q property. The CQT-derived spectrogram will then exhibit greater temporal resolution at higher frequencies than at lower frequencies.

Table 3.1: EERs (%) for the ASVspoof 2015 database, evaluation partition. Results for GMM-CQCC and GMM-LFCC baseline systems. Results for two attacks (S8 and S10) show stark differences in performance for each system.

System	S1	S2	S3	S4	S5	S6	S7	**S8**	S9	**S10**	Avg.
GMM-CQCC	0.01	0.11	0.00	0.00	0.13	0.09	0.06	**1.03**	0.05	**1.07**	0.26
GMM-LFCC	0.03	0.41	0.00	0.00	0.11	0.15	0.01	**0.07**	0.03	**8.19**	0.89

3.2 Performance on ASVspoof 2015 database

Both CQCC and LFCC front-ends were typically used with a classic Gaussian mixture model (GMM) classifiers. The performance obtained with GMM-CQCC and GMM-LFCC systems in terms of EER for the evaluation partition of the ASVspoof 2015 database is illustrated in Table 3.1, which reproduces results from [71]. Averages for all attacks (S1–S10) are shown in the last column. Results for two attacks show marked differences in performance for the CQCC and LFCC front-ends. These are S8 (a tensor-based voice conversion [119]), for which LFCCs outperform CQCCs by 93% relative, and S10 (unit-selection based TTS [1]) for which CQCCs outperform LFCCs by 87% relative. However, our assumption is for S10 attack as we will see later in Section 3.6.1, the artefacts are present at higher frequencies and CQCC-resampling [71] front-end which gives more emphasis to information at higher frequency, hence why CQCCs outperforms LFCCs for this attack. The comparatively high EER for S10 dominates, and so the average EER for all attacks is lower for CQCCs than for LFCCs. While the motivation for our work stems from these two stark differences for the ASVspoof 2015 database, results reported later in this chapter relate to the more recent ASVspoof 2019 database [49].

3.3 Experimental setup

Experiments were conducted using the standard ASVspoof 2019 LA database described in the Section 2.1. All experiments were conducted with the evaluation set only. Performance is assessed in terms of the EER [120] also as described in Section 2.2. Although the tandem detection cost function (t-DCF) metric [121] is the default metric for the ASVspoof 2019 challenge, the work reported in this chapter is focusing on CM performance rather than its impact on the ASV performance. We used the ASVspoof 2019 official challenge baseline systems provided by the organisers, namely the GMM-CQCC and GMM-LFCC systems. The CQCC features are extracted using 12-bins per octave, followed by re-sampling applied with a sampling rate of 16 kHz. The CQCC feature representation includes 19 static

[1] http://mary.dfki.de/

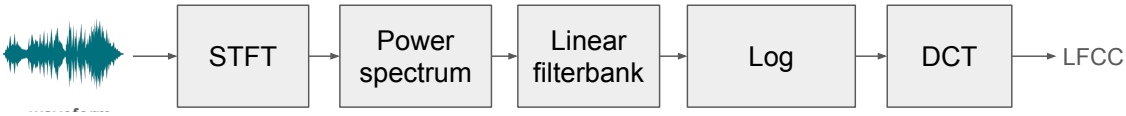


Figure 3.2: Block diagram of LFCC feature extraction.

with energy coefficients, velocity (Δ), acceleration ($\Delta\Delta$) coefficients. The LFCC features are extracted using 20 ms window length with a 10 ms frame-shift using a 512 -point FFT. The LFCC features comprise 20 static, Δ, and $\Delta\Delta$ coefficients, resulting in 60-dimension feature vectors. The feature extraction process for both baseline systems is illustrated in Figure 3.1 and Figure 3.2, respectively. Results reported in Table 3.2 are those obtained with the original ASVspoof 2019 baseline CMs as reported in [49]. Both baseline systems use a traditional GMM classifier with two models, one for bona fide speech and one for spoofed speech, each with 512 Gaussian mixture models. Output scores are conventional log-likelihood ratios.

3.4 Sub-band analysis

The variations in the spectro-temporal resolution of the DFT and CQT can potentially explain the differences in the performance of GMM-LFCC and GMM-CQCC systems. This is because the spectral resolution at lower frequencies and temporal resolution at higher frequencies are different. It may suggest that the artefacts which distinguish spoofed speech from bona fide speech may be located in specific sub-bands, rather than in the full-band signal. Previous research works [85, 122–124] also suggest that not all sub-bands are equally useful for detecting spoofing attacks. The artefacts present in certain frequency sub-bands may be more informative. This led us to investigate whether differences at the sub-band level could explain the difference in performance for the GMM-LFCC and GMM-CQCC systems. The set of experiments conducted to test this hypothesis consists of an extensive sub-band analysis in which the GMM-LFCC and GMM-CQCC classifiers are applied to the ASVspoof 2019 LA database at a sub-band level. In each experiment, the entire database is processed with a low-pass and/or high-pass filter. With both low-pass and high-pass filters, the result is a band-pass filter with cut-in f_{min} and cut-off f_{max}. Corresponding GMM models are retrained each time to learn band-specific features. The filter cut-in and cut-off frequencies are varied in steps of 400 Hz between 0 Hz and the sampling frequency of 8 kHz.

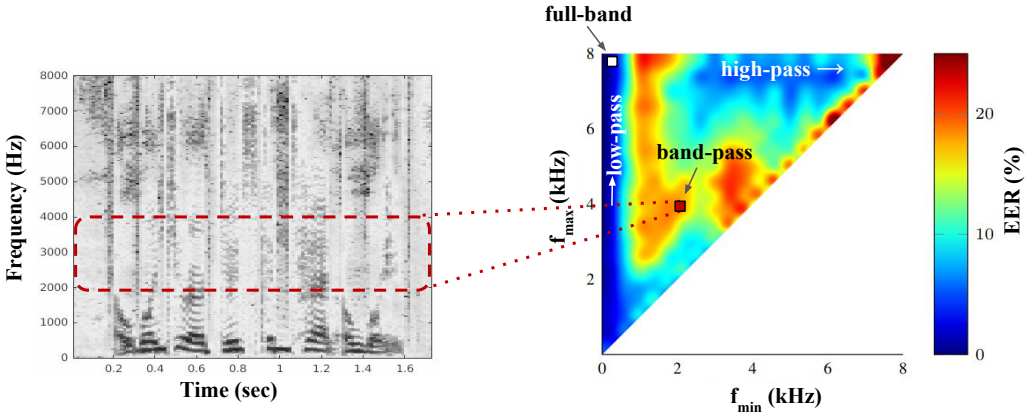


Figure 3.3: (a) Audio spectrogram of a signal; (b) A 2-dimensional heatmap visualisation of sub-band analysis results for an arbitrary spoofing attack. The horizontal axis depicts the high-pass cut-in frequency f_{min} whereas the vertical axis depicts the low pass cut-off frequency f_{max}. The colour bar to the right of the plot depicts the EER obtained for each band-pass filter configuration (pair of f_{min} and f_{max}). (Best viewed in colour.)

3.5 Heatmap visualisation

We apply a 2-dimensional heatmap, a visualisation method to LFCC and CQCC front-end features to show where the artefacts are localised in the spectrum. Sub-band results are visualised in the form of a 2-D heatmap representations for specific spoofing attacks, as exemplified in Figure 3.3. We use a low-pass, high-pass and band-pass filter to generate a sub-band signal such as that highlighted in Figure 3.3-(a). The horizontal and vertical axes of the heatmap represent the cut-in frequency f_{min} and cut-off frequency f_{max} of the high-pass and low-pass filters, respectively. For band-pass filters, $f_{\text{min}} < f_{\text{max}}$, hence the triangular form of the heatmap in Figure 3.3-(b) . Each point in the 2-D heatmap represents the EER of a given sub-band with different cut-in and cut-off frequency pairs. The EER corresponding to each sub-band configuration is indicated with the colorbar to the right of Figure 3.3-(b), where blue colours indicate lower EERs and red colours indicate higher EERs. The left-most column of the heatmap shows the EER with no high pass filtering and an increasingly aggressive low pass filtering moving from top to bottom. The top-most row of the heatmap shows the EER with no low pass filtering and an increasingly aggressive high-pass filter moving from left-to-right. The full-band configuration is located at the top-left. Everywhere else corresponds to a band-pass filter. EERs along the diagonal of Fig-

ure 3.3-(b) highlight the significance of spectrum information at the sub-band level.

For the arbitrary example shown in Figure 3.3, EERs along the diagonal suggest that information at lower frequencies is discriminative, whereas information at higher frequencies is not as discriminative when used alone. EERs in the left-most column reveal that the most discriminative information is found at low frequencies. As long as this information is used, then the EER is low. EERs in the top-most row demonstrate that, as soon as low frequency information is discarded, then the EER increases. Information between 1 and 3 kHz and above 7 kHz is not discriminative. Information between 3 and 7 kHz is less discriminative, and reasonable EERs can only be achieved when the information from different sub-bands is combined.

3.6 Results

This section describes the sub-band analysis results for ASVspoof 2015 and 2019 LA databases.

3.6.1 Sub-band results for ASVspoof 2015 database

Results for the CQCC-GMM and LFCC-GMM CMs in Table 3.1 show substantial variations in performance for S8 and S10 attacks. CQCC-GMM works well for the S10 attack and LFCC-GMM works well for the S8 attack. To explain this, we performed sub-band analysis experiments for S8 and S10 attacks and the result are

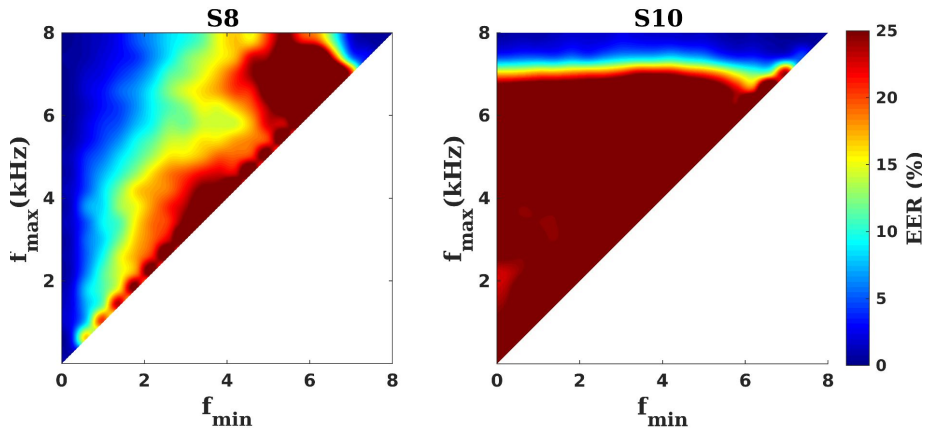


Figure 3.4: Sub-band analysis results; (a) 2-D heatmap visualisation for S8 attack with LFCC-GMM system and (b) 2-D heatmap visualisation for S10 attack with CQCC-GMM CM. (Best viewed in colour.)

illustrated in Figure 3.4. As shown in Figure 3.4-(a) the artefacts for S8 attack are localised in the lower to a higher frequency in the spectrum those easily captured by DFT derived LFCC front-end features as it gives constant resolution to all frequency bands in the spectrum whereas CQCC-resampling front-end emphasis more on higher frequency might fail to capture artefacts localised in the lower frequency. On the other hand for the S10 attack as shown in Figure 3.4-(b) the artefacts are localised in the higher frequencies, at which CQCC-resampling front-end gives more emphasis to information, hence why CQCCs outperforms LFCCs. Even only using information between 7.8 and 8 kHz frequency band as shown in Figure 3.4-(b), CQCC-GMM CM can easily obtain lower EER.

3.6.2 Sub-band results for ASVspoof 2019 LA database

EER results for the two CMs assessed using the evaluation partition of the ASVspoof 2019 LA database are illustrated in Table 3.2. They show substantial variations in performance for the GMM-LFCC and GMM-CQCC CMs. For attacks A07, A16 and A19, the GMM-CQCC system outperforms the GMM-LFCC system, whereas for attacks A13, A14 and A17, it is the GMM-LFCC system which performs best, although still with relatively high error rates. Subsequent experiments were performed separately for this subset of 6 specific spoofing attacks, all examples of where one front-end representation leads to substantially better results than the other. Brief details of the specific algorithms used in creating each spoofing attack are summarised in Chapter 2 (see Table 2.2). Among them are four voice conversion algorithms (A13, A14, A17 and A19) and two speech synthesis algorithms (A07 and A16), even though the input to two of the voice conversion algorithms is also synthetic speech (A13 and A14). Full details of each attack algorithm are available in [53].

Sub-band analysis results for these six attacks are shown in Figure 3.5. Results for the GMM-CQCC and GMM-LFCC systems are illustrated in rows one and two, respectively. In each case, the EER for the full-band systems is indicated by the top-left-most point in each 2D heatmap. Figures 3.5 (a)-(c) show that the baseline GMM-CQCC system (top-left-most point) gives low EERs, whereas Figures 3.5 (g)-(i) show that the GMM-LFCC system performs consistently worse. Figures 3.5 (d)-(f) and (j)-(l) show the opposite, even if the performance for A17 (l) is still poor for both systems. Our analysis of the heatmaps for both CQCC and LFCC front-ends reveal that the discriminative information for the detection of all three attacks is located at higher frequencies. This suggests that both CMs rely heavily on information extracted from the higher-frequency band. While not visible in the plots at this scale, the discriminative information lies above 7.6 kHz; near-to-zero EERs can be obtained using information between 7.6 and 8 kHz only,

Table 3.2: EERs (%) for the ASVspoof 2019 logical access database, evaluation partition. Results for both baseline systems, GMM-CQCC (linear scale), GMM-LFCC and GMM-CQCC (geometric scale). Results for six attacks (highlighted) show stark differences in performance for each system.

System	**A07**	A08	A09	A10	A11	A12	**A13**	**A14**	A15	**A16**	**A17**	A18	**A19**
GMM-CQCC (resampling)	**0.00**	0.04	0.14	15.16	0.08	4.74	**26.15**	**10.85**	1.26	**0.00**	**19.62**	3.81	**0.04**
GMM-LFCC	**12.86**	0.37	0.00	18.97	0.12	4.92	**9.57**	**1.22**	2.22	**6.31**	**7.71**	3.58	**13.94**
GMM-CQCC (geometric)	**3.39**	0.34	0.46	6.86	4.62	3.58	**4.23**	**0.67**	1.52	**4.00**	**25.04**	19.63	**29.46**

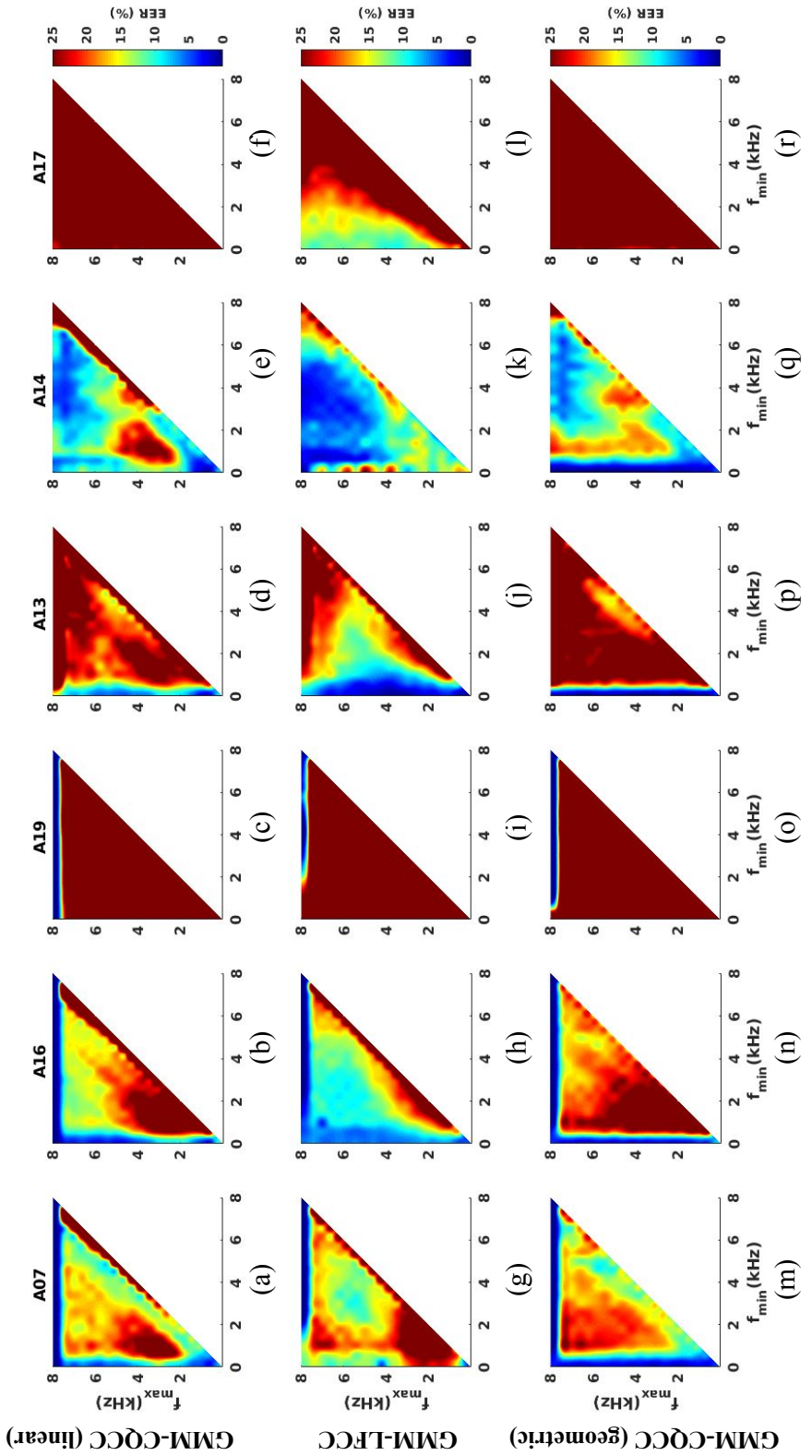


Figure 3.5: 2-D heatmap visualisations of sub-band analysis results for the six different spoof attacks. The first row (a-f) shows results for the GMM-CQCC (linear) system. The second row (g-l) shows corresponding results for the GMM-LFCC system. The third row (m-r) shows results for the GMM-CQCC (geometric) system.

using either front-end. However, the situation for attacks A13, A14 and A17 is slightly different. The diagonals of the heatmaps in Figures 3.5 (d), (e), (j) and (k) all suggest that the most informative information is at lower frequencies. The left-most columns of Figures 3.5 (d), (e) and (j) suggest that the most critical information is located at the very lowest sub-bands for the LFCC front-end. For A17, neither front-end performs especially well, with the CQCC front-end failing completely, whereas the LFCC front-end is able to capture information across the majority of the full band.

3.7 Spectro-temporal resolution

The results presented above dispel somewhat our hypothesis that CQCCs perform better than LFCCs for some attacks due to their higher temporal resolution at high frequencies. If this were true, reliable performance would not have been obtained with the LFCC front-end using high-frequency sub-bands alone. The explanation lies elsewhere. The same results show that the discriminative information simply lies in the highest sub-bands; with appropriate band-pass filtering, both front-ends perform well for attacks A07, A16, and A19. However, this straightforward explanation does not account for why the original, *full-band* CQCC front-end also performs well, i.e. *without* band-pass filtering. The explanation for this observation requires us to revisit the issue of spectro-temporal resolution in CQT and DFT-based front-ends.

Figure 3.6 illustrates a set of DFT and CQT-derived spectra for an arbitrary speech frame. Each plot also shows the second basis function of the DCT (black solid line) used in cepstral analysis. The vertical bars serve to illustrate the *spectral* resolution across the spectrum and give some indication of the *temporal* resolution. It is inversely proportional to the spectral resolution. The top plot in Figure 3.6 shows the CQT-derived spectrum (without re-sampling) [125]. It clearly shows that the spectral resolution is higher at lower frequencies than at higher frequencies. The DCT basis function is plotted with a logarithmic frequency scale, hence the regularity of the vertical lines and DCT basis function. These plots demonstrate that, without resampling, the DCT will attribute greater *emphasis* to lower frequency components and generate a smoother spectrum (orange shaded area in the top plot) than to higher frequency components. The second plot shows the CQT-derived spectrum after resampling to a uniform frequency scale. The vertical bars in Figure 3.6 show how resampling acts to linearise the sampling rate in the frequency domain. Note the difference in frequency scales for the top and middle plots. The higher spectral resolution (and hence lower temporal resolution) for lower frequencies is still clearly apparent; the spectrum is much smoother at higher frequencies (light green shaded area in the

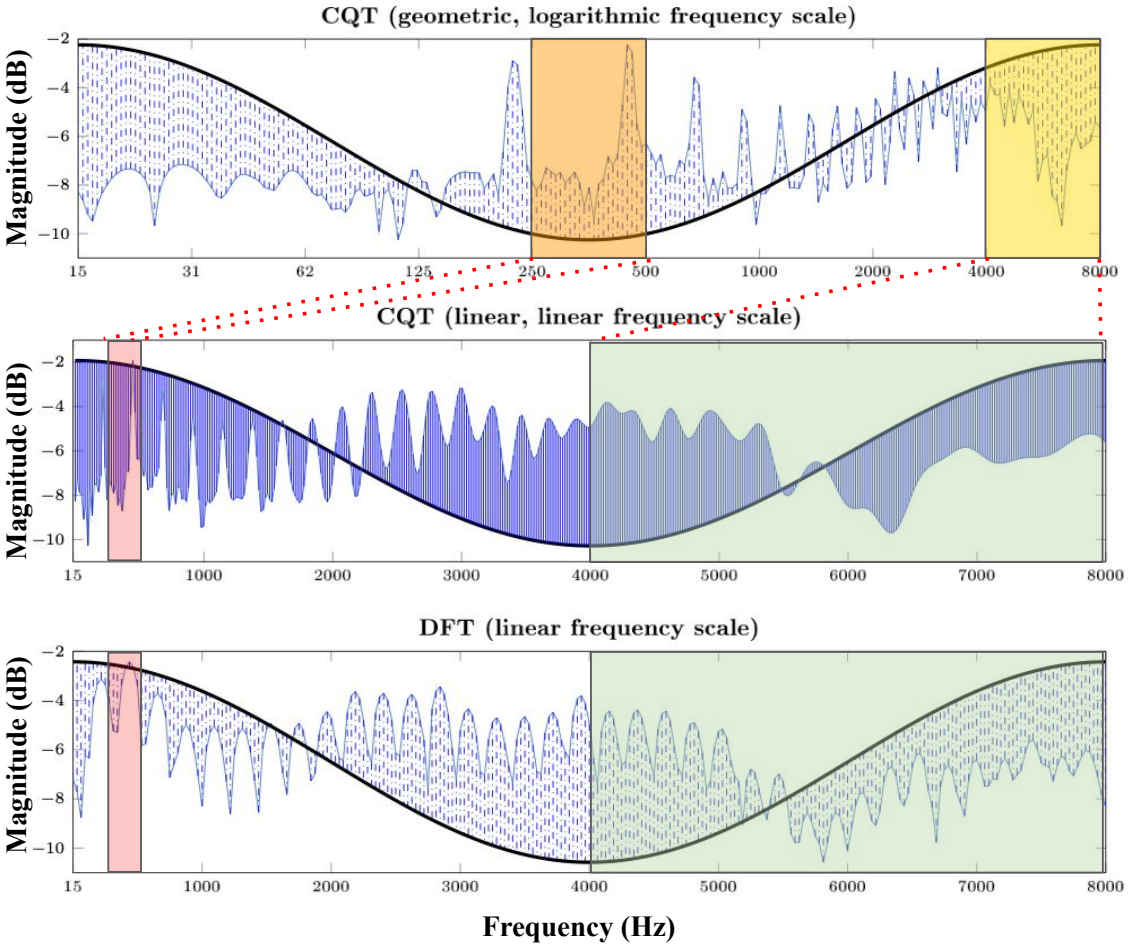


Figure 3.6: Illustrations of spectra for an arbitrary speech frame derived using the CQT and DFT (blue) and the second basis function of the discrete cosine transform used in cepstral analysis (black solid line). The top plot shows the CQT-derived spectrum without resampling, hence the logarithmic sampling (compression of vertical blue lines) in the logarithmic frequency scale. The second plot shows the CQT-derived spectrum after resampling to a uniform, linear frequency scale (Section 2; thus more vertical lines). The third plot shows the corresponding DFT-derived spectrum. (Best viewed in color.)

middle plot) after resampling.

A comparison of the cosine basis functions to the CQT-derived spectra in the top and middle plots shows that resampling acts to dilute (weaken) information at lower frequencies (pink shaded area) but to distil (emphasise) information at higher

frequencies (light green shaded area in middle plot) compared to the CQT-dervied spectrum without resampling (yellow shaded area in top plot). Hence, spoofing artefacts localised at high frequencies are emphasised by CQCC-resampling. The bottom plot of Figure 3.6 shows the DFT-derived spectrum and linear sampling in the frequency domain. Here, the DCT acts to weight all frequency components uniformly. Spoofing artefacts at high frequencies are not clearly emphasised; reliable performance is then obtained only by using band-pass filtering. The explanation for why the LFCC front-end outperforms the CQCC front-end for attacks A13, A14 and A17 is now straightforward. The artefacts for these specific attacks are at lower and mid-range frequencies where the CQT acts to dilute information, hence why the GMM-CQCC system performs poorly for these attacks. Since the DFT gives uniform weighting to information at all frequencies in the spectrum, the GMM-LFCC system gives better, though still somewhat poor performance. This is because information at low frequencies is not *emphasised*.

3.8 Validation

To validate these findings, we performed an additional experiment with the original CQCC (geometrically-scaled) front-end. The CQT-derived spectra in the top plot of Figure 3.6 show more dense sampling of the spectrum at lower frequencies but sparse sampling at higher frequencies. Hence, without resampling, the cepstral analysis (DCT) should emphasis the information situated at lower frequencies. In this case, a CQCC front-end *without* linear resampling should produce better detection performance in the case of attacks A13 and A14 for which artefacts are located at lower frequencies. Performance for attack A17 might still not be improve, since the artefacts appear not to be localised especially at lower frequencies. We repeated the sub-band analysis experiments described in Section 3.4 using the geometrically-scaled CQCC front-end.

Results are illustrated in Figures 3.5 (m)-(r) for the same set of 6 attacks for which EER results are shown in the last row of Table 3.2. The comparison of the right-most columns in the heatmaps of Figures 3.5 (d) and (p) and those in (e) and (q) clearly show that lower EERs are obtained using the geometrically-scaled CQCC, rather than the linearly-scaled CQCC front-end. Low EERs are even obtained when the front-end is applied without any band-pass filtering. This is because the original CQT geometric frequency scale results in the emphasis of information at lower frequencies where the artefacts are located. As expected, attack A17 remains troublesome. For some attacks, the EER for the geometrically-scaled CQCC front-end is slightly higher, but is slightly lower for others. Once again, performance for A17 is poor and even worse than for both the linearly-scaled CQCC and LFCC front-ends. Those for A13 and A14 clearly confirm our findings,

with substantially lower EERs than those obtained with both the linearly-scaled CQCC and LFCC front-ends.

3.9 Summary

This chapter presents an explanation for why the CQCC front-end works so reliably in detecting some forms of spoofing attack, but why it fails to generalise to others. Through the sub-band analysis of CQCC-GMM CM system, we observe that discriminative information in the spectrum is present at the different sub-band, not all frequency sub-bands are informative. This confirms that different spoofing attacks exhibit artefacts at different frequencies in the spectrum, artefacts that are better captured with specific front-ends. The standard CQT exhibits a dense sampling of the spectrum at lower frequencies and a sparse sampling at higher frequencies. Hence, geometrically-sampled CQCCs perform well in detecting spoofing artefacts when they are located at low frequencies. Linear sampling (resampling) shifts the emphasis to higher frequencies so that artefacts at similarly high frequencies are emphasised and captured reliably. The main findings from this chapter are that no single CQCC or LFCC front-end configuration can perform well for diverse range of spoofing attacks; different sub-band front-ends have different potential to distinguish spoofed speech from bona fide. These findings may explain why classifier fusion has proven to be so important to CM generalisation, i.e. reliable performance in the face of varying spoofing attacks. We may then need to rethink the approach to classifier fusion to exploit the complementary strengths of each CM.

Chapter 4

A Non-linear Ensemble of Sub-band Countermeasures

In Chapter 3, we have seen that no single front-end can effectively detect all types of spoofing attacks. This finding is confirmed by ASVspoof 2019 challenge results which have consistently shown that the most reliable performance in detecting spoofed speech requires the use of an ensemble of different front-ends [47, 83, 126]. Designing a single robust model to detect unforeseen attacks can be challenging, as also shown by ASVspoof 2019 challenge results [47]. The top-performing system used an ensemble of seven sub-systems. As evident from the literature [85, 124, 127, 128] and Chapter 3 findings, we argue that not all frequency bands are useful for spoof detection. The work presented in the Chapter 3 showed that spoofing artefacts lie at the sub-band level and have better potential be to captured by front-ends with appropriate spectral resolution within the same frequency band.

To this end, we propose in this chapter an ensemble of sub-band countermeasures (CMs), each tuned to detect artefacts in different sub-bands. We will see that it can improve spoofing detection performance beyond what can be achieved through the fusion of CMs operating directly upon the full-band signals. However, our assumption that different front-ends are required to detect artefacts within different sub-bands implies that traditional linear approaches to score fusion may not be optimal. This is because a single spoofing attack may only be detected reliably by a single CM within an ensemble. In this case, linear approaches to score fusion may not fully utilise the complementary strengths of each CM, as they may simply combine mostly non-informative scores while diluting any informative scores. Non-linear approaches to fusion may hence have better potential to exploit the complimentarity of different sub-band CMs.

The key contributions of this work include:

- introduce a optimised version of the LFCC-GMM baseline using high-spectral resolution front-end features.
- propose a non-linear ensemble of sub-band countermeasures to leverage attack-specific discriminative information present in the different sub-bands.

4.1 Research hypotheses

To better illustrate the ideas explored in this chapter, we consider the hypothetical anti-spoofing example illustrated in Figure 4.1. Plotted on each axis are the scores produced by two different spoofing CMs: CM_1 and CM_2. CM_1 is tuned to detect artefacts present within a lower frequency band, at 0-4 kHz for example. CM_2 is tuned to detect artefacts within a higher frequency band, at 4-8 kHz for example. Each point in the 2D plot signifies the scores produced

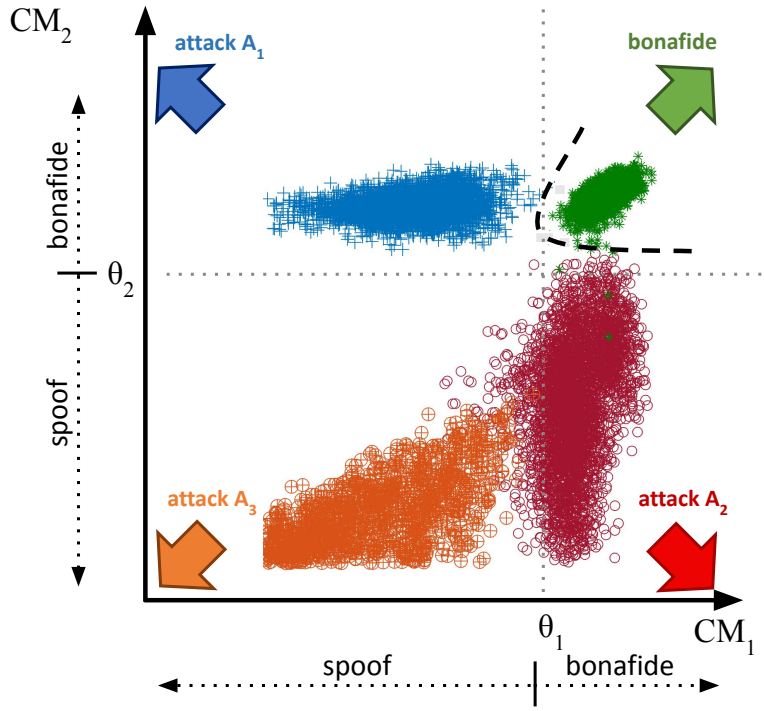


Figure 4.1: A scatter plot of scores for CM_1 and CM_2. Clusters correspond to bona fide utterances (green) and three spoofing attacks (A1-blue, A2-red, A3-orange). The dashed black line indicates a non-linear decision boundary that best separates bona fide from spoofed utterances.

by each CM for a set of bona fide and spoofed utterances. Scores for bona fide (genuine) utterances are illustrated by green points (top-right). The scores for three different types of spoofing attacks are also shown: Attack A_1, characterised by artefacts predominantly at low frequencies (blue points, top-left); Attack A_2, characterised by artefacts at high frequencies (red points, bottom right); Attack A_3, which exhibits artefacts at both low and high frequencies (orange points, bottom left). The first hypothesis is that different spoofing attacks are characterised by artefacts within different sub-bands in the spectrum and that an ensemble of different front-ends are needed in order to detect such artefacts reliably.

Both CMs produce predominantly high scores for bona fide utterances; as per standard ASVspoof practice, high CM scores reflect bona fide trials, whereas low scores reflect spoof trials. Since CM_1 and CM_2 and their respective thresholds θ_1 and θ_2 are tuned for the detection of spoofing attacks A_1 and A_2 respectively, spoofing attack A_1 provokes mostly low scores for CM_1 and mostly high scores for CM_2, and vice versa for attack A_2. Attack A_3 provokes low scores for both CMs. Considering *multiple diverse attacks and CMs*, a notional decision boundary that best separates bona fide from spoofing utterances might correspond to a non-linear function. Linear score fusion operators may not perform well in this case, leading to poor reliability. The second hypothesis is that a non-linear approach to score fusion or system combination is needed in order to best exploit the complementarity of an ensemble of CMs tuned for the detection of specific spoofing attacks.

4.2 Sub-band countermeasures

In Chapter 3, we used two different CQCC front-ends that were tuned to increase spectral resolution at either low or high frequencies. The main idea in this chapter is an ensemble of sub-band CMs, each of which is tuned for the detection of a specific set of spoofing attacks and the associated artefacts, regardless of their location in the spectrum. Since the CQT has a non-linear spectral resolution [117, 129] that is difficult to tune to specific sub-bands, we adapted the LFCC front-end, which has a linear spectral resolution and can easily be applied at the sub-band level. Here, we describe the strategy of spectral resolution and front-end tuning at the sub-band level for the detection of specific spoofing attacks and present the results for each front-end when used with a GMM back-end and tested against each unseen spoofing attack in the ASVspoof 2019 LA database [49] described in Section 2.1.

Table 4.1: min t-DCF, EER and Bhattacharyya distance between bona fide and spoofed score distributions for different numbers of sub-band filters N. Baseline configuration illustrated in bold; selected configuration in italics.

Filters (N)	min t-DCF	EER (%)	D_B
20	**0.2110**	**2.71**	**0.1338**
30	0.0000	0.79	0.1706
40	0.0000	0.00	0.1770
50	0.0000	0.00	0.1785
60	0.0000	0.00	0.1793
70	*0.0000*	*0.00*	*0.1826*
80	0.0000	0.00	0.1788
90	0.0000	0.00	0.1823
100	0.0000	0.00	0.1830
120	0.0000	0.00	0.1820

4.2.1 High-spectral resolution front-end

Our findings in Chapter 3 showed that reliable spoofing attack detection requires a front-end with higher spectral resolution within the relevant frequency band. However, using a spectral resolution that is too high can result in noisy features. Hence, before sub-band optimisation, we set out first to optimise the spectral resolution at the full-band level. Sub-band optimisation was performed using the full ASVspoof 2019 LA training and development subsets [49]. While other techniques could also have been applied, e.g. zero padding, larger window-sizes, we simply modified the baseline LFCC front-end to use 30 ms frame blocking with a 15 ms frame-shift and used a larger 1024-point Fourier transform. The resolution was then decreased using a filterbank in the usual fashion by varying the number of filters N [123]. Higher values of N capture greater spectral detail, while the use of fewer filters may cause over-smoothing in the spectrum [130] and loss of critical spectral detail resulting in degraded classification performance. Hence, it is necessary to find the optimal number of filters for reliable spoof detection.

Results depicted in Table 4.1 show CM performance in terms of the min t-DCF and EER against the number of filterbank filters N (first 3 columns). For $N > 30$ filters, both the min t-DCF and the EER are zero. An alternative approach to optimisation is hence necessary. We elected arbitrarily to use the Bhattacharyya

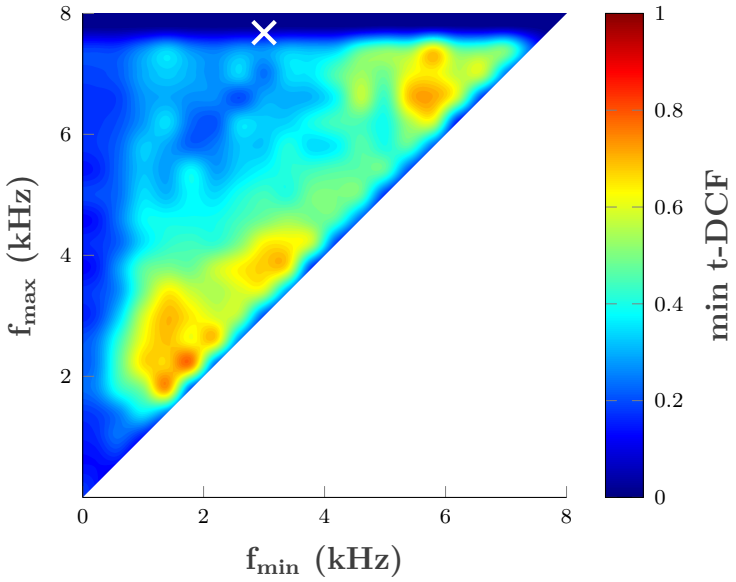


Figure 4.2: A 2-D heatmap visualisation illustrating sub-band level CM performance for attack A04 of the ASVspoof 2019 LA database. The cut-in frequencies f_{min} and cut-off frequencies f_{max} are indicated on horizontal and vertical axes respectively. Those of the CoM-defined sub-band is indicated by the white cross.

distance [131] between the CM score distributions for bona fide and spoofed trials. The Bhattacharyya distance is a simple metric to compute the distance between two Gaussian distributions by computing the mean and variances of two separate classes or distributions. The greater the distance between score distributions the better the separation between classes. The Bhattacharyya distance between bona fide and spoof Gaussian distributions is defined by Eq.(4.1):

$$D_B(b,s) = \frac{1}{4} ln \left(\frac{1}{4} \left(\frac{\sigma_b^2}{\sigma_s^2} + \frac{\sigma_s^2}{\sigma_b^2} + 2 \right) \right) + \frac{1}{4} \left(\frac{(\mu_b - \mu_s)^2}{\sigma_b^2 + \sigma_s^2} \right) \tag{4.1}$$

where subscripts b and s indicate parameters for bona fide and spoofed score distributions and where μ and σ refer to the means and standard deviations respectively. Results in the last column of Table 4.1 show that the distance between score distributions increases for $N > 30$ filters, but with little gain beyond $N = 70$ filters, which is the configuration used for all further experiments reported in this chapter.

4.2.2 Sub-band selection

Attack-specific, sub-band front-ends are designed using heat-map visualisations (see Section 3.5) which indicate CM performance at the sub-band level. An exam-

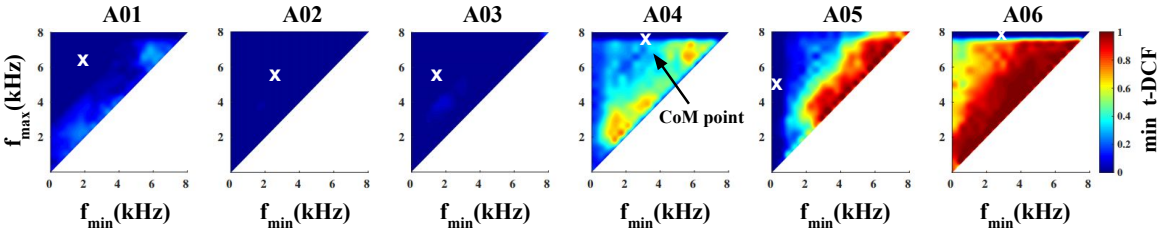


Figure 4.3: A 2D heatmap visualisation illustrating sub-band level CM performance for the six spoofing attacks (A01-A06) on the development set. The cut-in frequencies f_{min} and cut-off frequencies f_{max} are indicated on horizontal and vertical axes respectively. Each CoM-defined sub-band is indicated by the white cross in a heatmap. (Best viewed in colour.)

ple for the A04 attack is illustrated in Figure 4.2. The heat-map colour signifies CM performance in terms of min t-DCF for a front-end with cut-in and cut-off frequencies of f_{min} (x-axis) and f_{max} (y-axis) respectively. We used the Centre-of-Mass (CoM) approach [132] to identify a single point in the heat-map to define a specific sub-band for the detection of each attack in the development subset (A01–A06). The CoM is a crude means of coping with a noisy surface containing multiple minima. The CoM of a distribution of mass in space is the unique point where the weighted relative position of the distributed mass sums to zero. We consider the 2D heat-map as a system of particles P_i where $i = 1, \ldots, n$. Each particle has coordinates $r_i = [f^i_{\text{min}}, f^i_{\text{max}}]$ and mass $m_i = (\text{min t-DCF}_i)^{-1}$. The coordinates $R = [f^{CoM}_{\text{min}}, f^{CoM}_{\text{max}}]$ of the CoM satisfy the condition $\sum_{i=1}^{n} m_i(r_i - R) = 0$. Solving for R yields:

$$R = \frac{1}{M} \sum_{i=1}^{n} m_i r_i \tag{4.2}$$

where M is the sum of the masses of all the particles in the full 2D heat-map representation. We obtain a different R for each spoofing attack and hence define six attack-optimised, sub-band CMs for spoofing attacks in the ASVspoof 2019 LA development dataset. The CoM-defined sub-bands for each spoofing attacks (A01-A06) are defined by the white cross in Figure 4.3. They correspond to the bandwidth of each sub-band CM in first column of Table 4.2.

4.3 Non-linear fusion of sub-band classifiers

The second hypothesis under investigation in this chapter is that a non-linear approach to score fusion or system combination is needed in order to best exploit the complimentarity of an ensemble of CMs, each tuned for the detection of a specific spoofing attack. In order to validate this hypothesis, we explored four

different fusion techniques comprising both linear and non-linear approaches to effectively combine sub-band CM scores. They are defined as follows:

(i) **Support Vector Machine (SVM) based fusion**: A SVM is a supervised machine learning algorithm which uses a decision boundary known as a hyperplane, to separate data points into different classes [133]. The objective is to find the best hyperplane that maximises the margin between bona fide and spoofed classes. We used a SVM-based fusion approach to find the optimal decision boundary to distinguish between bona fide and spoofed class. A SVM model is trained on the input feature vector generated from a set of sub-band CM scores for the ASVspoof 2019 LA development set. We used a polynomial kernel function with a seventh-order polynomial. Linear and residual basis kernel functions were also tested, however, they yielded inferior results. The SVM model was then tested using evaluation set and the same sub-band CM systems to produce the fused output score. The polynomial kernel function represents the similarity between training samples in a feature space over polynomials of the original variables, allowing for the learning of a non-linear decision boundary [133].

(ii) **Gaussian Mixture Model (GMM) based fusion**: Another non-linear approach is the GMM-based fusion approach used in [134] to combine ASV and CM scores. Inspired by this work, we also explored non-linear GMM-based fusion with 64 Gaussian component models learned from the set of scores for bona fide and spoofed classes and compute log-likelihood ratios (LLRs) as the fused output score. We also tested models with 16, 32, and 128 GMM components, but these yielded inferior results.

(iii) **Multinomial Logistic Regression (MLR) fusion** [135] and (iv) **traditional linear fusion** using the bosaris toolkit [55]: Both approaches are supervised. Fusion weights are again optimised using ASVspoof 2019 LA development set. Both fusion methods linearly combine the outputs of multiple sub-systems into a single score and produce LLR outputs.

4.4 Results

Experiments were performed using the ASVspoof 2019 LA database [49] described in Section 2.1. We present results for the sub-band CMs, and fusion performance and compare their results to the ASVspoof 2019 challenge official baseline systems without any modifications, namely the GMM-CQCC (B1) and GMM-LFCC (B2) systems [49] as described in Section 3.3, as well as results for the top-performing ASVspoof 2019 LA challenge entries.

Table 4.2: Results in terms of min t-DCF for development (A1-A6) and evaluation (A07-A19) partitions. P1 and P2 indicates a pooled min t-DCF and pooled EER on the evaluation subset. Results in boldface signify the attack for which each sub-band is optimised, e.g. the CM designed for attack A01 operates within a sub-band of 2011 to 6403 Hz.

Subband	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A07**	**A08**	**A09**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**P1**	**P2**
2k-6.4k	**.00**	.00	.00	.22	.25	.79	.37	.03	.00	.55	.03	.18	.31	.17	.15	.25	.41	.60	.93	.34	13.3
2.4k-5.6k	.00	**.00**	.00	.31	.42	.91	.35	.06	.00	.54	.08	.23	.31	.16	.16	.33	.58	.87	.99	.39	15.5
2k-5.6k	.00	.00	**.00**	.27	.31	.82	.37	.06	.00	.56	.08	.22	.28	.18	.17	.30	.48	.81	.99	.38	14.7
3.2k-8k	.00	.00	.00	**.00**	.15	.00	.00	.00	.00	.54	.00	.41	.71	.10	.23	.00	.47	.29	.00	.26	10.6
.015k-4.8k	.00	.00	.00	.16	**.00**	.51	.45	.01	.00	.54	.07	.09	.09	.12	.13	.16	.33	.60	.85	.31	12.3
3.6k-8k	.00	.00	.00	.00	.12	**.00**	.00	.00	.00	.55	.00	.40	.79	.09	.31	.00	.55	.24	.00	.27	11.6
0-8k	.00	.00	.00	.00	.00	.00	.00	.00	.00	.15	.00	.11	.07	.06	.06	.00	.35	.07	.00	**.09**	**3.50**

4.4.1 Sub-band countermeasures

Results presented in Table 4.2 show the performance of six single sub-band systems (rows 2-7) and one full-band CM (row 8) in terms of the min t-DCF [56]. Columns P1 and P2 show performance in terms of the pooled min t-DCF and EER (%), respectively for the evaluation set. The bandwidth of each CoM defined sub-band CM is illustrated in the first column (in kHz). Results for the development set (columns A01-A06) show that sub-band CMs all yield zero min t-DCFs for the attacks they were designed for (results in boldface), as they also do for some other attacks. This is not surprising given the considerable spectrum overlap among the set of sub-band CMs. Interestingly, the full-band CM is the only one to achieve zero min t-DCF for all six attacks in the development set. Results for the evaluation set (columns A07-A19) show that the full-band CM gives similar or lower min t-DCFs than individual sub-band CMs. These observations are confirmed by pooled min t-DCFs for the evaluation set. These results raise questions about: (i) whether or not the ensemble of attack-specific, sub-band CM scores can give better performance when their individual performance is poor relative to the full-band CM performance; (ii) what should be the best fusion approach to exploit the complimentarity of an ensemble of sub-band CMs tuned for the detection of specific spoofing attacks.

4.4.2 Fusion

Table 4.3 presents the performance of all four fusion methods in terms of the min t-DCF. Columns P1 and P2 show pooled min t-DCF and pooled EER results for the evaluation set respectively. Results for the development set (columns A01-A06) show that all fusion techniques achieve zero min t-DCFs for all spoofing attacks. The non-linear GMM-based fusion approach achieves the best performance, with a pooled min t-DCF of 0.074 for the evaluation set. The next best system is the non-linear SVM fusion approach with a min t-DCF of 0.075. The two linear approaches yield t-DCFs of 0.091 and 0.118. These findings seem to confirm the hypothesis that a non-linear approach is better suited to the fusion of sub-band CMs. This is because spoofing artefacts that are localised in the spectrum may be detected only by sub-band CMs whose focus is directed towards the same parts of the spectrum and hence be detected reliably by a subset of CMs only (or even only a single CM). In this case, full-band CMs may dilute relevant information by smoothing across the spectrum and linear approaches to fusion may not identify the best decision boundary between bona fide and spoofed speech. Better performance is obtained with non-linear decision boundaries.

Table 4.3: Fusion results in terms of min t-DCF for development (A1-A6) and evaluation (A07-A19) partitions. Columns P1 and P2 indicates a pooled min t-DCF and pooled EER on the evaluation subset. Fusion system indicates the combination of all single subband systems with full-band system using different linear and non-linear fusion techniques. MLR: multinomial logistic regression.

Sys.	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A07**	**A08**	**A09**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**P1**	**P2**
GMM	.00	.00	.00	.00	.00	.00	.00	.00	.00	.16	.00	.07	.07	.02	.03	.00	.27	.07	.00	**.074**	**2.92**
SVM	.00	.00	.00	.00	.00	.00	.00	.00	.00	.20	.00	.08	.09	.02	.04	.00	.27	.08	.00	.075	2.92
Linear	.00	.00	.00	.00	.00	.00	.00	.00	.00	.21	.00	.08	.09	.02	.04	.00	.28	.11	.00	.091	3.38
MLR	.00	.00	.00	.00	.00	.00	.00	.00	.00	.30	.00	.12	.16	.07	.11	.00	.31	.09	.00	.118	4.50

Table 4.4: Performance for the ASVspoof 2019 evaluation partition in terms of pooled min t-DCF and pooled EER (%) for top-performing systems (T05, T45, T60 and T24), fusion results (in boldface) and baseline results (B1, B2).

System	min-tDCF	EER (%)
T05	0.0069	0.22
T45 [83]	0.0510	1.86
GMM fusion	**0.0740**	**2.92**
SVM fusion (polynomial kernel)	**0.0748**	2.92
T60 [126]	0.0755	2.64
High res. LFCC (Proposed single system)	0.0900	3.50
Linear fusion	**0.0910**	3.38
T24	0.0953	3.45
Multinomial logistic regression fusion	**0.1180**	4.50
Best single system in Challenge [47] (T45)	0.1560	5.06
LFCC:B2 [49]	0.2116	8.09
CQCC:B1 [49]	0.2366	9.57

4.4.3 Performance comparison

Table 4.4 also shows results for the two ASVspoof 2019 baseline systems (B1 and B2, last two rows) and the four top-performing (out of 48 submissions) challenge results [47,49]. The latter are denoted by their anonymous ASVspoof 2019 identifiers T05, T45, T60 and T24 [49]. Only T45 [83] and T60 [126] system descriptions are publicly available. It is worth noting that all four of these competing systems are based upon an ensemble of comparatively complex neural network-based architectures [47], in contrast to the very simple GMM-based solution used in our work. The top performing system (T05) used a combination of seven complex neural network-based sub-models, including MobileNet, DenseNet and different variant of ResNet architecture (see Figure 2 in [47]) to generate fused scores. Furthermore, they used a combination of *multiple, different* front-end parameterisations, unlike the *single, common* base front-end used in our work. As shown in Figure 4.4, even though the comparison is between evaluation and post-evaluation results, both non-linear GMM and SVM-based approaches to fusion outperform all but two of the 48 competing systems. Even though the gap is not substantial, the two non-linear approaches to fusion are outperformed by the traditional linear approaches. Our proposed single system, utilising high-spectral resolution based LFCC front-end and a simple GMM back-end, demonstrates a relative improvement of 42% in terms of min t-DCF by utilising only 10% of the training data compared to the best single system T45 (row 11 in Table 4.4).

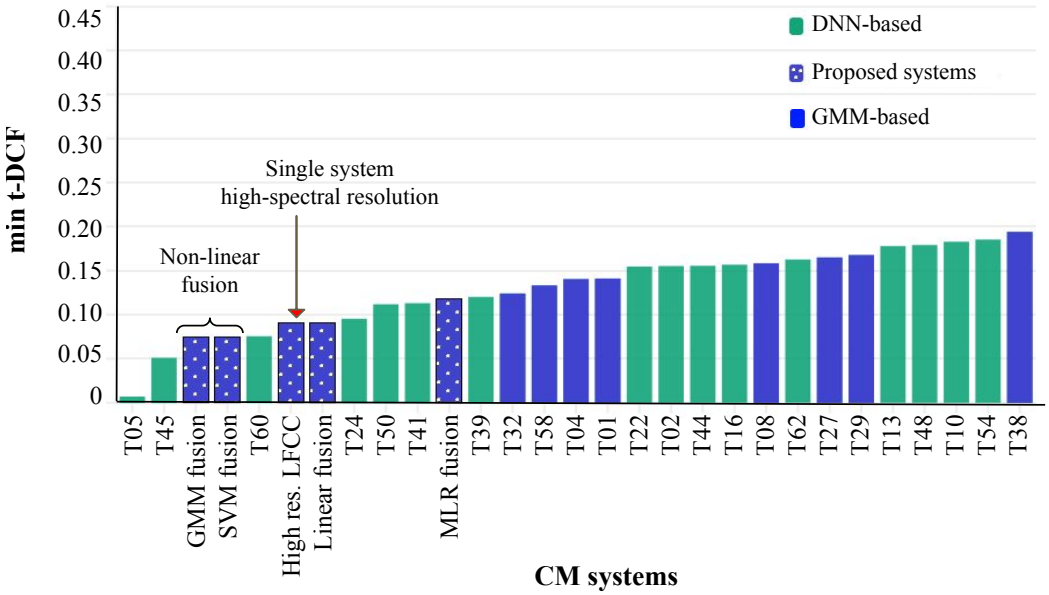


Figure 4.4: Performance comparisons with the ASVspoof 2019 LA challenge submissions.

4.5 Summary

This chapter investigates whether spoofing attacks leave sub-band artefacts that necessitate specific spoofing CMs for detection. It also explored the potential of a non-linear ensemble approach to effectively combine the set of scores produced by sub-band CMs. The competitive results obtained using a high spectral-resolution, full-band CM alone demonstrate the effectiveness of the front-end features. This finding could be beneficial to other anti-spoofing researchers that use neural networks with standard, low-resolution front-ends. Switching to high-resolution front-ends may improve detection performance. The proposed high-resolution front-end outperforms the challenge LFCC-GMM baseline by a large margin. Furthermore, the non-linear fusion of sub-band CMs improves detection performance showing the effectiveness of the non-linear fusion approach.

Chapter 5

End-to-End Anti-Spoofing Using RawNet2

For nearly a decade, the combination of a front-end feature extractor and a back-end classifier has been the standard framework for spoof detection. Within this framework, the majority of the existing works have mainly focused on the development of hand-crafted acoustic front-end features. The design of appropriate feature representations and suitable classifiers have often been considered as separate tasks. One drawback of these standard frameworks is that the designed features might not be optimal for the classification objective at hand. Since neural networks have the capability of learning representative features automatically, it raises the question of whether the use of handcrafted features is necessary or beneficial. The use of hand-crafted features can lead to the loss of useful information, such as phase information. To avoid this, why not allow the neural network to learn more discriminative feature representations automatically from the raw waveform. End-to-end frameworks operate directly upon raw waveform inputs, streamlining the training and evaluation process. The objective of this work is to develop more efficient, robust end-to-end spoofing and deepfake detection CM that can capture the fundamental differences between spoofed and bona fide speech, i.e., between machine-generated and human-generated speech.

Chapter 4 presented the high-resolution front-ends as a promising solution, but highlighted poor reliability for the worst-case A17 attack detection. A similar trend was seen for the top-performing entries of the ASVspoof 2019 LA challenge [47]. This may be because A17 attack exhibits more temporally-distinct artefacts, which are not ease captured by pre-processed hand-crafted features commonly used in conventional approaches. This highlights the need for a more sophisticated CM solution capable of detecting a wide range of previously unseen spoofing attacks. Rather than relying heavily on pre-processed, hand-crafted

acoustic features, it may be more beneficial to use end-to-end (E2E) neural network model that can automatically learn the relevant information in both the temporal and spectral domains for reliable spoofing detection. This work is motivated by literature analysing raw waveform-based DNNs front-end [136], as well as the success of E2E models for related speech processing tasks, such as speech recognition [137], speech separation [138], and speaker verification [139]. To the best of our knowledge, this is the first successful application of end-to-end RawNet2 network for anti-spoofing and audio deepfake detection.

The key contributions of this work include:

- introduce a robust E2E deep neural network for anti-spoofing and deepfake detection;
- motivated by Chapter 3, we incorporate different spectral resolutions (mel, linear, inverse-mel spaced filters) in the sinc-layer front-end, in order to effectively capture artefacts that are localised in different frequency-bands;
- improve the worst case scenario A17 spoofing attack detection performance by a large margin compared to the state-of-the-art solutions at the time of publication.

5.1 Related work

This section provides a brief overview of previous work which led to the development of RawNet2 [140]. In recent years, E2E classifiers have been widely adopted in speech processing tasks, where every component between and including any DNN-based front-end and back-end classification is jointly optimised [137, 140, 141]. Many of these solutions which operate directly upon the raw waveform inputs, thus avoiding the limitations imposed by the use of knowledge-based, hand-crafted acoustic features. There is already a significant amount of research in the automatic speaker verification (ASV) literature which reports E2E solutions operate directly upon raw waveform. One of the early solutions is RawNet [140], introduced in 2019. RawNet is a convolutional neural network architecture that generates speaker embeddings for the verification task. The first convolutional layer is applied directly to the raw waveform input, with all filter parameters being learned automatically to extract representative features for the given task. Among the higher layers are deep residual neural networks [79] which extract frame-level representations. They use either long short-term memory (LSTM) as in [139] or gated recurrent units (GRUs) as in [140], to aggregate utterance-level representations and either a b-vector classifier [142, 143] as in [139] or a simple DNN classifier with concatenation and multiplication (concat & mul)

scoring approach as used in [140] for speaker verification. The use of a wholly unconstrained convolutional neural network (CNN) layer whose parameters are learned automatically can increase the number of network parameters resulting in slow convergence. Additionally, the first layer outputs also tend to be noisy, especially when training data is limited.

One approach to address these issues is the E2E SincNet [144] network proposed in 2018 for speaker recognition. While the higher layers of the SincNet architecture are relatively standard, the first convolutional layer operates on the raw waveform and is composed of a bank of band-pass filters parameterised in the form of sinc functions. Sinc filters act as time-invariant rectangular band-pass filters. The use of a constrained first layer, with fewer learnable parameters, whereby only the cut-off frequencies and filter bandwidths are learned with a fixed rectangular-shaped filter response, leads to the learning of a more meaningful filterbank structure and outputs. With fewer parameters, the convergence speed for SincNet is faster than a standard convolutional layer.

An improved version of RawNet [145] (RawNet2) combines the merits of the RawNet [140] approach with the use of SincNet as a front-end to learn representative features for speaker verification. The first layer of RawNet2 is essentially the same as that of SincNet, whereas the upper layers consist of the series of six residual blocks and GRU layers. Additionally, RawNet2 incorporates filter-wise feature map scaling (FMS) using a sigmoid function which is applied individually to each filter output. Despite the growing interest in E2E approaches, only one study in the literature explored E2E approach for anti-spoofing [146]. In 2017, dinkel et al. proposed the first raw waveform-based E2E convolutional long short-term neural network (CLDNN) [146]. The CLDNN model uses CNN and LSTM layers as a front-end to extract a utterance-level representations and the DNN as back-end classifier. However, the CLDNN model was evaluated on other, older databases (BTAS [147] and AVspoof [148]) with a half total error rate (HTER) evaluation metric, making it difficult to compare its performance to that of other models. The recent speaker recognition literature [140, 141] suggest that E2E architectures which avoid the use of hand-crafted features have potential to improve upon performance. We sought to determine whether the benefit of E2E automatic feature learning translates well to anti-spoofing, especially in the worst-case scenario A17 attack [47].

5.2 RawNet2 architecture for anti-spoofing

In this section, we describe modifications made to the original RawNet2 architecture [145] in order to effectively adapt it to the anti-spoofing and deepfake

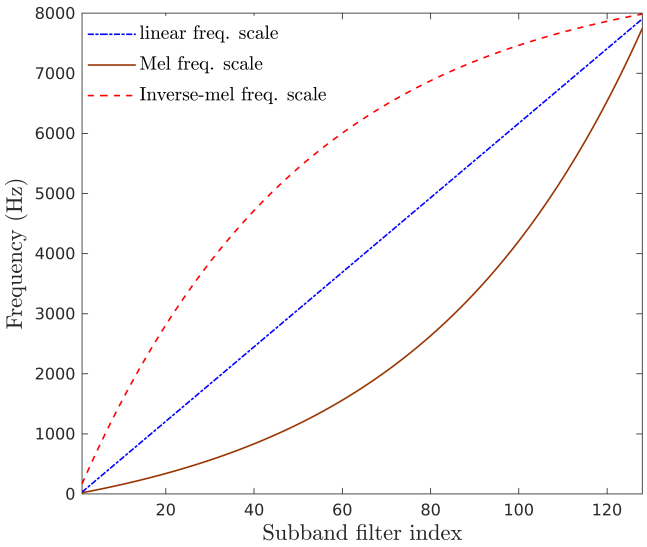


Figure 5.1: Spectral resolutions (frequency-scale) used in sinc-layer front-end initialisation.

detection task. Details of these modifications can be found in Table 5.1, in which changes to parameters or architecture components are highlighted in bold text. The first modification concerns the first layer of the architecture, which ingests raw speech. Since it leads to worse performance, we did not apply any layer normalisation [149] to the raw input. On account of training data sparsity, or rather the limited number of different spoofing attacks (only 6 for the training & development set for the ASVspoof 2019 LA database), we neither learn cut-off frequencies nor the bandwidth of each sinc filter. Other experiments, not reported here showed that learning cut-off frequencies leads to over-fitting. To prevent over-fitting, we neither learn automatically the bandwidth nor spectral position of each sinc filter.

Inspired by the success of different filterbank spectral resolutions as described in [16] and Chapter 3, we also initialised sinc filters with mel-distributed, linearly-distributed and inverse-mel spaced filters, as shown in Figure 5.1. We optimised the filter length (number of filter coefficients). This is because the duration of cues used to detect spoofing are not necessarily the same as those for speaker recognition. We use a series of six residual blocks [79] with pre-activation [150] to extract the high-level feature representation. Each residual block comprises a 1D convolutional layer, batch normalisation [151] with SeLU activation units [152], and a max-pooling layer for data down-sampling.

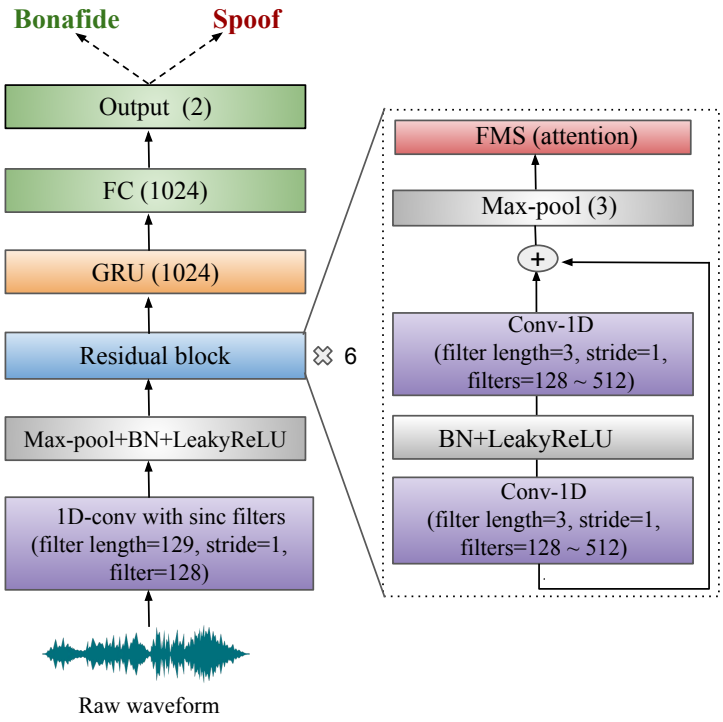


Figure 5.2: End-to-end RawNet2 framework.

We apply FMS independently to the output of each residual-block in the network. FMS acts as an attention mechanism [153] applied in the spectral-domain. It acts to emphasise the most informative filter outputs by assigning higher attention weights to that filter. To conduct FMS, we first perform global average pooling operation on the residual-block output to obtain the feature-map corresponding to each filter. Then, a scalar vector is derived by feed-forwarding each filter output through a fully-connected linear layer followed by a sigmoid layer. We adopt a combined additive and multiplicative feature scaling approach to obtain a scaled feature-map output as described in [145]. A GRU layer [154] with 1024 hidden nodes is used to aggregate frame-level representations into a single utterance-level representation. Instead of producing speaker embeddings as in RawNet2 [145], the GRU output is followed by an additional fully connected layer which precedes the output layer. Output layer is then applied in order to produce two-class predictions: bona fide or spoofed. The architecture is illustrated in Figure 5.2, whereas the full architecture is summarised in Table 5.1.

Table 5.1: The RawNet2 architecture used for anti-spoofing. Modifications made to the original architecture are highlighted in boldface. BN refers to batch normalisation.

Layer	Input≈**64000** samples	Output shape
Sinc filters	Conv(**129**,1,128) Maxpooling(3) BN & LeakyReLU	(21290,128)
Residual-block	{ BN & LeakyReLU Conv(3,1,128) BN & LeakyReLU Conv(3,1,128) Maxpooling(3) FMS } × 2	(2365,128)
Residual-block	{ BN & LeakyReLU Conv(3,1, **512**) BN & LeakyReLU Conv(3,1, **512**) Maxpooling(3) FMS } × 4	(29, 512)
GRU	GRU(1024)	(1024)
FC	1024	(1024)
Output	**1024**	**2**

5.3 Experiments

Experiments were performed on the ASVspoof 2019 LA database described in Section 2.1. There was no standard E2E anti-spoofing system available at the time of publication. Therefore, we use the high-spectral resolution-based LFCC-GMM CM system presented in Chapter 4 as a baseline. It uses 30 ms windows, a 15 ms frame-shift and 70 linearly-spaced filters with conventional cepstral analysis and a simple GMM back-end classifier. RawNet2[1] is implemented using PyTorch, a deep learning toolkit in Python. The first sinc layer is initialised with 128 mel-scaled filters and a filter size of 129 samples (8 ms duration) which is convolved with the raw waveform. Audio waveforms are truncated or concatenated to give segments of 4 second (64000 samples) duration. The entire architecture is trained using the ASVspoof 2019 LA training set to minimise a weighted cross entropy (WCE) loss function, where the ratio of weights assigned to bona fide and spoofed trials are 9:1 to manage the data imbalance in the training set. The standard Adam optimiser [155] is used with a mini-batch size of 32, and a learning rate of 0.0001. The network is trained for 100 epochs. The feature extractor and back-end

[1] https://github.com/eurecom-asp/rawnet2-antispoofing

classifier are jointly optimised using back-propagation [156]. The best model was selected based on the minimum validation loss on ASVspoof 2019 LA development set.

5.4 Results

Results are reported for three different RawNet2 variants, their fusion performance. Performance comparisons are made to the top-performing ASVspoof 2019 LA challenge entries [47, 49]. Results are presented in terms of the minimum tandem detection cost function (min t-DCF) [56] and the equal error rate (EER) described in Section 2.2.

RawNet2

Table 5.2 illustrates the results for the baseline (L) and three different RawNet2 variants (S1-S3) for the ASVspoof 2019 LA evaluation set. Columns P1 and P2 show performance in terms of the pooled min t-DCF and EER (%), respectively. The baseline exhibits a pooled min t-DCF of 0.090, whereas the best performing RawNet2-S2 system, which uses fixed inverse Mel-spaced sinc filters gives a best pooled min t-DCF of 0.1175. Results for the A17 attack, which is known to be the most difficult attack for the baseline and other top-performing ASVspoof 2019 LA challenge entries [47], is more favourable for the RawNet2-S3 system which uses linearly-spaced sinc filters, and for the RawNet2-S1 system which uses Mel-spaced sinc filters. The baseline system gives a min t-DCF of 0.3524 for attack A17, whereas the RawNet2-S3 system exhibits a lower min t-DCF of 0.181 (49% relative reduction in min t-DCF). To the best of our knowledge, this represents one of the best EER results reported for the A17 attack at the time of publication.

Fusion

Since the high resolution LFCC-GMM baseline system provides the best overall pooled result, and the RawNet2-S3 system performs best for the worst case A17 attack, it is worth evaluating the benefits of their fusion to exploit their complementarity. Experiments were conducted using the support vector machine (SVM) based fusion approach (see Section 4.3). A SVM-based approach used to fuse the high-spectral resolution baseline (L) with the different RawNet2 variants such as RawNet2-S1, RawNet2-S2 and RawNet2-S3. Table 5.3 presents the fusion performance. The results for the two official AVSspoof 2019 challenge baseline systems: the CQCC-GMM (B1) and the low-spectral resolution LFCC-GMM (B2) baseline [49], are also shown in the Table 5.3. All systems outperform the baselines, with the high-spectral resolution baseline (L) giving a min t-DCF of 0.0904, marginally better than the min t-DCF of 0.0953 for team T24. Team T60 produced a min t-DCF of 0.0755, and team T45 achieved a min t-DCF of 0.0510. All fusions

Table 5.2: Results in terms of min t-DCF for evaluation (A07-A19) partition and respective pooled min t-DCF (P1) and pooled EER (P2). S1: fixed Mel-scaled sinc filters. S2: fixed inverse Mel-scaled sinc filters. S3: fixed linear-scale sinc filters.

System	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	P1	P2
LFCC-GMM	.001	.001	.000	.154	.005	.115	.080	.070	.069	.001	.352	.074	.008	**0.090**	**3.50**
RawNet2-S1	.028	.198	.012	.037	.019	.038	.043	.019	.031	.046	.262	.615	.053	0.130	5.64
RawNet2-S2	.013	.055	.013	.030	.010	.036	.008	.005	.034	.019	.263	.624	.041	0.117	5.13
RawNet2-S3	.038	.111	.035	.049	.050	.044	.046	.038	.038	.046	**.181**	.528	.068	0.129	4.66

Table 5.3: Performance for the ASVspoof 2019 evaluation partition in terms of pooled min t-DCF and pooled EER for top-4 systems (T05, T45, T60 and T24), SVM-based fusions of high-spectral-resolution LFCC (L) and RawNet2 systems (boldface), and ASVspoof 2019 baseline systems (B1, B2). Individual min t-DCF results for A17 are illustrated in the right-most column.

System	min-tDCF	EER	worst case A17 (min t-DCF)
T05	0.0069	0.22	0.0040
L+S1	**0.0330**	**1.12**	**0.1161**
L+S1+S2+S3	**0.0347**	**1.14**	**0.0808**
L+S3	**0.0370**	**1.14**	**0.0965**
L+S2	**0.0443**	**1.35**	**0.1339**
T45 [83]	0.0510	1.86	0.2208
T60 [126]	0.0755	2.64	0.3254
L [157]	0.0904	3.50	0.3524
T24	0.0953	3.45	0.3547
LFCC:B2 [49]	0.2116	8.09	0.2880
CQCC:B1 [49]	0.2366	9.57	0.5859

of the high-spectral resolution baseline (L) with either S1, S2 and S3 RawNet2 systems perform better, with the L+S1 combination even outperforming fusion with all three RawNet2 variants (L+S1+S2+S3). The last column of Table 5.3 shows the performance in terms of the min t-DCF for the worst-case scenario A17 attack. Just like individual system results shown in Table 5.2, all RawNet2 fused systems produce a substantially lower min t-DCF than the baseline system (L), confirming that the end-to-end RawNet2 system is complementary as it learns artefacts that the baseline system fails to capture. Last, the results for team T05[2] show a lower pooled min t-DCF and lower min t-DCF for the worst-case scenario A17 attack, our results are fully reproducible with open-source code.

5.5 Summary

In this chapter, we present the successful application of an E2E RawNet2 to anti-spoofing and audio deepfake detection. Our results show that E2E automatic feature learning has a positive impact on spoofing detection, especially in the worst-case scenario, compared to traditional hand-crafted feature learning. Additionally, fusion results suggest that the RawNet2 classifier is capable of learning cues that are distinct from those captured by traditional front-end features. Tra-

[2]Note that the best ensemble T05 system remains unreproducible to date.

ditional acoustic features and the features learned directly from raw waveform are hence complementary, and combining them can lead to further improvements in performance. E2E modes are widely known for their ease of training and minimal need for feature engineering, which makes them efficient and practical for various application. Moreover, the RawNet2 model has been adopted as one of the strong baselines for the ASVspoof 2021 challenge.

Chapter 6

End-to-End Spectro-temporal Graph Attention Network

The work presented in Chapter 5 demonstrates that a model can be optimised to learn the relative importance of specific spectral and temporal intervals. This work is inspired by the work reported in Chapter 5 and the psychoacoustic studies [115, 116], which demonstrated the ability of the human auditory system to select the most discriminative spectral bins and to perform an auto-correlation between adjacent temporal frames. Therefore, the learning of correlation between spectral and temporal cues may improve spoof detection reliability. In recent years, graph neural networks (GNNs) [158] and their variants have seen an impressive level of success in learning the relationships among different features (nodes). Inspired by the power of GNNs to model complex relationships among different nodes/edges, we explored their use to model the relationship between the spectral and temporal representations. The self-attention mechanism [153] assigns weights to different nodes based on their relevance to neighboring nodes, with highly weighted node features likely contributing more to the final model prediction. The contribution in this chapter lies in proposing an end-to-end attention-based framework using GNNs to learn the relationship between discriminative spoofing cues for reliable detection.

6.1 Motivation

In previous chapters, we have learned that spoofing attacks often leave detectable cues in specific sub-bands or temporal intervals. A model can be trained to recognise the relative importance of both spectral and temporal spoofing cues. However, traditional methods for detecting these cues often rely on ensemble systems, each of which is specifically designed to detect a particular type of artefact

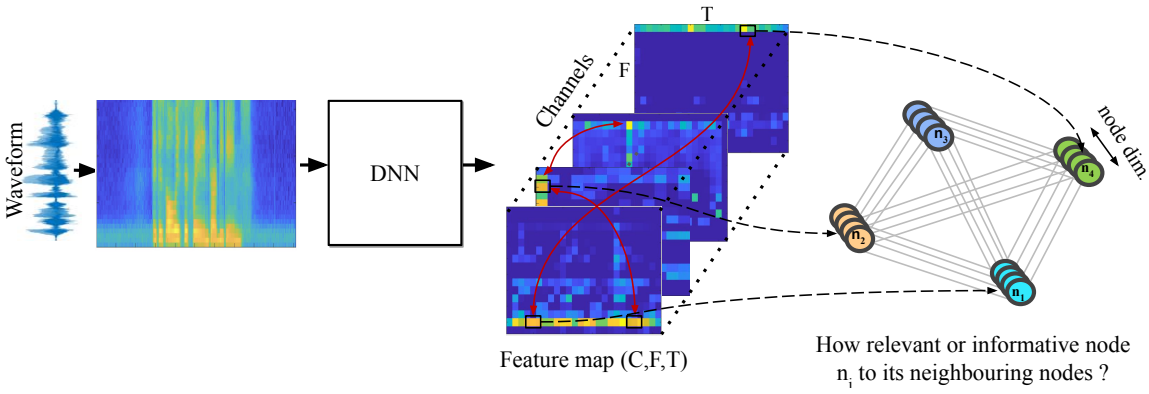


Figure 6.1: Illustration depicting the application of graph neural networks for spoof detection.

either in the spectral or temporal domain. This can be computationally expensive. To address this issue, we sought to learn the spectral and temporal artefacts simultaneously and model the relationship between them to improve detection using a single end-to-end system. To illustrate the concept of using GNNs for spoofing detection, we use high-level feature representations learned from a deep neural network in Figure 6.1. As shown in Figure 6.1, the discriminative information or spoofing cues can be localised in both spectral bands and temporal intervals. To capture this information efficiently, we need to learn the correlation between them.

To model the relationship between artefacts spanning different sub-bands and temporal intervals, we explored the use of graph attention networks (GATs) [159]. One of the main motivations for using GATs is their ability to effectively capture complex relationships between different features in the data. In the context of audio anti-spoofing, it is important to identify relationships between different feature representations, such as spectral and temporal characteristics, that may be indicative of spoofing. GATs are well-suited to this task as they are designed to learn and assign weights to dependencies between different features in the data using attention mechanisms. As shown in Figure 6.1 to construct the input graph, we consider each time-frequency (T-F) atom (single 'bin' across the time and frequency axis) as a node in a non-euclidean space, specifically, a graph. The number of channels represents the node feature dimensionality. The input graph generation process is summarised in Section 6.4.

Work presented in Chapter 5 showed the merit of attention mechanisms to capture artefacts located in different spectral sub-bands or temporal intervals. The objective in this chapter is to model the relationships using a joint spectro-temporal

attention network. This approach facilitates the aggregation of complementary, discriminative information concurrently in both domains. In this chapter, we introduce an end-to-end spectro-temporal graph attention network, called RawGAT-ST, which operates directly on the raw waveform. Such a fully end-to-end approach is designed to maximise the potential of capturing discriminative cues in both spectral and temporal domains concurrently. Inspired by the end-to-end speaker verification model [144, 145] and in building upon our end-to-end anti-spoofing solution, RawNet2, reported in Chapter 5, the proposed RawGAT-ST model uses a bank of sinc-shaped rectangular band-pass filters which operate directly upon the raw audio waveform through time-domain convolution. The key contributions of this work include:

- a fully end-to-end architecture comprising feature representation learning and graph modeling;
- a novel spectro-temporal graph attention network which learns the relationships between cues at different sub-band and temporal intervals;
- a new attentive graph pooling strategy to reduce computational cost and to improve discrimination power by discarding irrelevant nodes;
- the exploration of different model-level, graph combination strategies.

6.2 Related work

In recent years, GNNs [158, 160–163] have attracted growing attention, particularly its variants, such as graph convolution networks (GCNs) [164] and GATs [165]. GNNs are a type of neural network designed to operate on graph-structured data, such as the relationship between different segments of a speech signal. GNNs use an aggregation function to update the feature representation of each node by aggregating feature representations from neighboring nodes. Some commonly used aggregation functions include Mean, Max and Summation [163, 166]. GNNs also use a readout operation such as simple average or max-pooling to combine the feature information of all nodes into a single feature representation for the entire graph. Several prior studies have demonstrated the effectiveness of GNNs and their variants in various speech processing tasks [167–171], including audio classification, speaker verification and speaker diarization.

Zhang et al. [167] applied GCNs to a few-shot audio classification task in order to derive an attention vector that helps to improve the discrimination between different audio classes. Tzirakis et al. [170] used GCNs to find spatial correlations among the different microphone channels (considered as nodes in the graph)

for a multi-channel speech enhancement problem. The speech signal received from each microphone channel is first transformed into a time-frequency (T-F) representation, which is then fed into a neural network framework. They used a U-Net neural network architecture [172] where the encoder extracts high-level features from the inputs. The high-level feature representations obtained for each microphone channel are subsequently used to construct a input graph that captures the multi-channel information through its nodes and edges, and GCNs are used to aggregate information from each node (microphone channels). Jung et al. [169] demonstrated the use of GATs as a back-end classifier to learn the relationships between enrolment and test utterances for speaker verification. Kwon et al. [171] used GATs for multi-scale speaker diarisation to calculate the similarity between the two speech segments and for speaker clustering to identify the speakers. GATs can be used effectively to model the relationship between different segments of a speech signal, which can be useful for various speech processing tasks.

Our preliminaries study on GATs, as reported in [173], demonstrated how GATs can be used effectively to model the relationship between spoofing artefacts localised in different sub-bands and temporal frames. This is accomplished by utilising a self-attention mechanism [153], which focuses on the most salient sub-bands or temporal frames and the relationships between them. We applied GATs separately to model the relationships in either spectral or temporal domains and demonstrated their complementarity through a score-level fusion (i.e., late fusion). Our hypothesis is that the integration of these two graphs has better potential to leverage complementary information between spectral and temporal graph representations and to further enhance detection performance while using a *single*, end-to-end system.

6.3 Graph attention networks

A graph is defined as:

$$\mathcal{G}(N, \mathcal{E}, \mathrm{h}), \tag{6.1}$$

where $N = \{1, ..., n\}$ is the set of nodes, and $\mathcal{E}$ represents the edges between all possible node connections, including self-connections. We assume that every node has a feature vector $\mathrm{h} \in \mathbb{R}^d$, where d is the dimensionality of the feature vector. GNNs are a class of neural network which can model the non-Euclidean data manifold between different nodes by utilising high-dimensional feature vectors as the nodes. In this work, we consider *fully-connected* graphs with edges between every pair of nodes, including self-connections. Algorithm 1 provides an overview of the GAT learning process. The input graph $\mathcal{G}$ is formed from a set of node features, h , where $\mathrm{h} \in \mathbb{R}^{N \times d}$, where N is set of nodes and where d is the feature dimensionality (number of channels in the feature map). The GAT layer operates

Algorithm 1 The GAT Learning Process

Input: Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathrm{h})$, where $\mathrm{h}_n \in \mathbb{R}^d$ for $n \in \mathcal{N}$

Output: Updated graph $\mathcal{G}' = (\mathcal{N}, \mathcal{E}, \mathrm{o})$, where $\mathrm{o}_n \in \mathbb{R}^{d'}$ for $n \in \mathcal{N}$

for $n \in \mathcal{N}$ **do**

$\forall u \in \mathcal{M}(n) \cup \{n\}$, where $\mathcal{M}(n)$ is the set of neighboring nodes of n

$\alpha_{u,n} \leftarrow \mathrm{softmax}(W_{\mathrm{map}}(\mathrm{h}_n \odot \mathrm{h}_u))$

$m_n \leftarrow \sum_u \alpha_{u,n} \mathrm{h}_u$, node aggregation w.r.t Eq. 6.2

$\mathrm{o}_n \leftarrow \mathrm{SeLU}(\mathrm{BN}(W_{\mathrm{att}}(m_n) + W_{\mathrm{res}}(\mathrm{h}_n)))$

end

Output: Updated graph $\mathcal{G}' \leftarrow \mathrm{o}$

upon an input graph $\mathcal{G}$ to produce an output graph $\mathcal{G}'$. Steps involved in the GAT learning process described in the following sections.

6.3.1 Node aggregation

Each graph transmits information between its neighboring node pairs. During the node aggregation process, the information from a single node is gathered by aggregating information from its neighboring node pairs using learnable weights via a self-attention mechanism. Node features are aggregated with attention weights that reflect the connective strength or relationship between a given node pair. The information from neighboring nodes is aggregated according to:

$$m_n = \sum_{u \in \mathcal{M}(n) \cup \{n\}} \alpha_{u,n} \mathrm{h}_u, \tag{6.2}$$

where $\mathcal{M}(n)$ refers to the set of neighboring nodes for node n, and $\alpha_{u,n}$ refers to the attention weight between nodes u and n. The GAT layer assigns learnable attention weights α to each edge (see Figure 6.5). The attention weights are calculated using a feed-forward neural network and the softmax [174] function, which allows individual nodes to aggregate discriminative information.

6.3.2 Self-attention mechanism

The self-attention mechanism is an effective information aggregator [159]. Since different neighboring nodes may contain different information, we use the self-attention mechanism to learn the importance of each node w.r.t. its neighboring nodes. Attention weights reflect how relevant or informative one node is to another,

where higher weights imply stronger correlations between node pairs. In order to update the node features, a shared learnable linear transformation is applied to each node. Then a self-attention mechanism is applied via a single feed-forward neural network. The attention weight α between each neighboring node pair is derived according to:

$$\alpha_{u,n} = \frac{\exp(W_{\text{map}}(\mathrm{h}_n \odot \mathrm{h}_u))}{\sum_{w \in \mathcal{M}(n) \cup \{n\}} \exp(W_{\text{map}}(\mathrm{h}_n \odot \mathrm{h}_w))}, \tag{6.3}$$

where $W_{\text{map}} \in \mathbb{R}^{d' \times 1}$ is the learnable attention weights multiplied to the dot product of every neighboring node pair $h_n \in R^{d'}$ and $h_u \in R^{d'}$, and where $\odot$ denotes element-wise multiplication. The estimated attention weight $\alpha_{u,n}$ indicates the importance of node u (neighboring node) to node n, which is then normalised across all nodes using the softmax function.

6.3.3 Output graph

The output graph $\mathcal{G}'$ comprises a new set of node features o_n, where $n \in N$, with target dimensionality $d' < d$. Output node feature o_n is derived according to:

$$\mathrm{o}_n = \text{SeLU}(\text{BN}(W_{\text{att}}(m_n) + W_{\text{res}}(\mathrm{h}_n))), \tag{6.4}$$

where SeLU refers to a scaled exponential linear unit [152] activation function, BN refers to batch normalisation [175], m_n is the aggregated information for node n, and $\mathrm{h}_n \in \mathbb{R}^d$ represents the feature vector of node $n \in N$. Each output node $\mathrm{o}_n \in \mathbb{R}^{d'}$ has a target dimensionality d'. $W_{\text{att}} \in \mathbb{R}^{d' \times d}$ is a learnable weight matrix which projects the aggregated information for each node n to the target dimensionality d' using a linear transformation layer. W_{res} projects the residual (skip) connection output to the same dimension. In this step, we obtain the final feature vectors for all nodes $n \in N$.

6.4 Spectro-temporal graph attention network

In this section, we introduce the proposed RawGAT model with spectro-temporal attention. It comprises four stages: i) learning higher-level semantic feature representations in truly end-to-end fashion by operating on the raw waveform (RawNet2 encoder); ii) a novel graph attention module with spectro-temporal attention; iii) a new graph pooling layer for discriminative node selection; iv) model-level fusion. The RawGAT-ST architecture is illustrated in Figure 6.2.

6.4.1 RawNet2 encoder

The RawGAT-ST model operates directly upon the raw waveform inputs. Some literature [140, 144, 176] shows that solutions based upon a bank of sinc filters

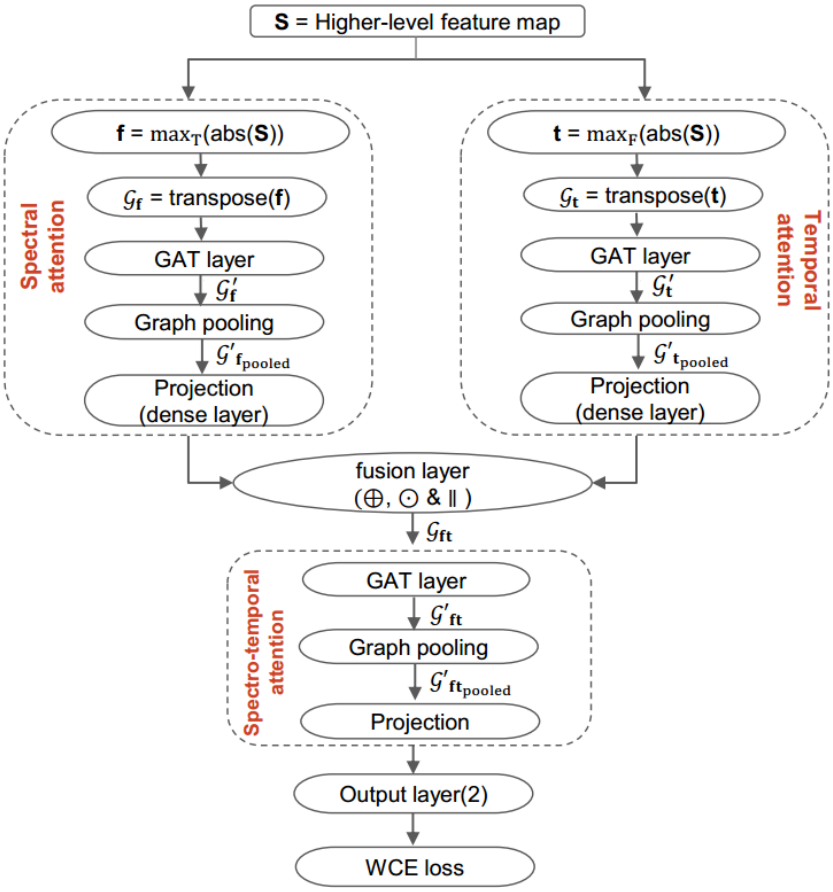


Figure 6.2: The proposed RawGAT-ST model architecture. The top-left spectral attention block captures discriminative spectral information. The top-right temporal attention block captures discriminative temporal information. Fusion is performed at the model level (middle ellipse). The bottom block comprises the spectro-temporal graph attention model.

are particularly effective in terms of both convergence stability and performance and also require fewer parameters compared to conventional convolution layers. Accordingly, we use a sinc convolution layer to learn feature representations similarly to our work in [176] and also reported in Chapter 5. It performs time-domain convolution of the raw waveform with a set of parameterised sinc functions which correspond to rectangular band-pass filters [122, 123]. The center frequencies of each sinc filter are initialised according to a mel-scale in an identical fashion to SincNet [144].

The choice of using sinc filters over conventional convolution filters is due to con-

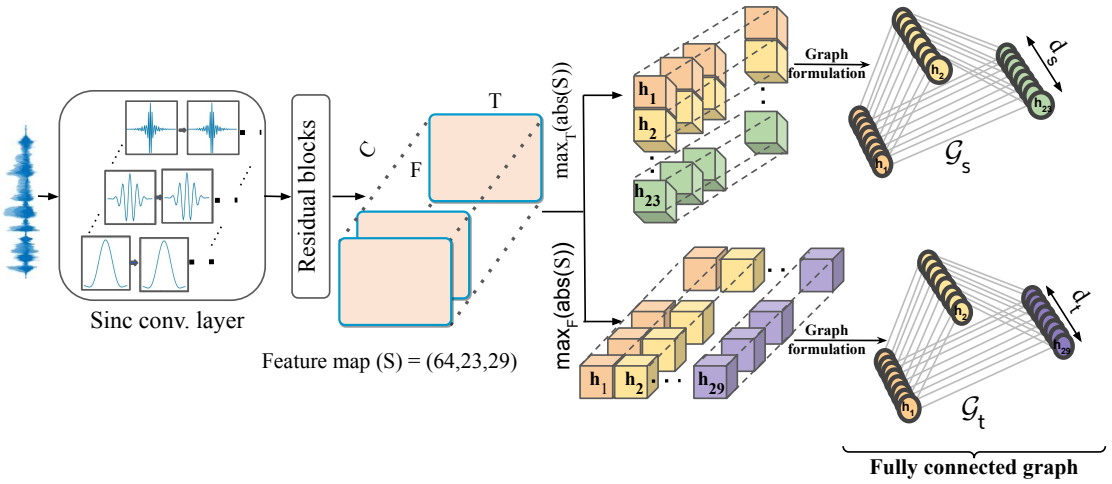


Figure 6.3: Illustration of graph formulation using higher-level feature representations.

strained parameterisation and fewer network parameters. In contrast to the original RawNet2 model, we interpret the output of the sinc layer as a 2-D time-frequency (T-F) representation by adding a single channel dimension, rather than a 1-D sequence. These 2-D features fed to a residual blocks [79] to extract the high-level representation $S \in \mathbb{R}^{C \times F \times T}$, where C, F and T refers to the number of channels, frequency bins and time samples respectively. Each residual block comprises a 2-D convolutional layer, a batch normalisation [151], SeLU activation unit [152] and a max-pooling layer. As illustrated in Figure 6.3, the higher-level feature extractor is identical to that of the RawNet2 encoder presented in Chapter 5.

6.4.2 Spectro-temporal graph attention

The approach to bring spectro-temporal graph attention to a single end-to-end model is a core contribution in this work. An overview of the proposed RawGAT-ST architecture is illustrated in Figure 6.2. The input to the model is a higher-level feature map (S) (top of Figure 6.2). Figure 6.3 depicts the temporal and spectral graph formulation from the higher-dimensional feature representation. The RawGAT-ST model comprises three principal blocks, each of which contains a single GAT layer: a spectral attention block (top-left of Figure 6.2); a temporal attention block (top-right); a final spectro-temporal attention block (bottom). The spectral and temporal attention blocks have the goal of emphasising the spectral and temporal cues. The third, joint spectro-temporal attention block

operates upon the pair of resulting graphs to model the relationships spanning both domains. All three blocks contain GAT [159] and graph pooling layers [177].

This process is first applied separately with attention to either spectral or temporal domains. Both spectral and temporal blocks operate upon the higher-level feature map $\mathrm{S} \in \mathbb{R}^{C \times F \times T}$. The spectral and temporal attention blocks first collapse temporal and spectral information respectively to a single dimension via max-pooling operation before the GAT layer. For the spectral attention block, max-pooling is applied to the absolute values across the temporal dimension thereby giving a spectral feature map $\mathrm{f} \in \mathbb{R}^{C \times F}$:

$$\mathrm{f} = \max_{\mathrm{T}}(\mathrm{abs}(S)), \tag{6.5}$$

The temporal attention block operates instead across the spectral dimension giving a temporal feature map $\mathrm{t} \in \mathbb{R}^{C \times T}$:

$$\mathrm{t} = \max_{\mathrm{F}}(\mathrm{abs}(S)), \tag{6.6}$$

Since S is derived from temporal data and hence contains both positive and negative values, use of absolute values in Eqs. 6.5 and 6.6 prevents meaningful negative-valued data from being discarded. Graphs $\mathcal{G}_{\mathbf{f}} \in \mathbb{R}^{N_{\mathbf{f}} \times d}$ and $\mathcal{G}_{\mathbf{t}} \in \mathbb{R}^{N_{\mathbf{t}} \times d}$ are then constructed from the transpose of f and t feature maps, respectively. $\mathcal{G}_{\mathbf{f}}$ contains a set of 23 nodes (the number of spectral-bins) whereas $\mathcal{G}_{\mathbf{t}}$ contains a set of 29 nodes (the number of temporal segments). Both graphs have a common dimensionality of $d = 64$ (number of channels). Separate GAT layers are then applied to both $\mathcal{G}_{\mathbf{f}}$ and $\mathcal{G}_{\mathbf{t}}$ to model the relationships between different sub-bands and temporal segments thereby producing a pair of new output graphs $\mathcal{G}'_{\mathbf{f}} \in \mathbb{R}^{N_{\mathbf{f}} \times d'}$ and $\mathcal{G}'_{\mathbf{t}} \in \mathbb{R}^{N_{\mathbf{t}} \times d'}$ with a common, reduced target dimensionality $d' = 32$.

6.4.3 Graph pooling

Various graph pooling layers have been proposed to obtain effective and discriminative graph representations [177, 178]. Graph pooling layers utilises learnable scoring functions to prune nodes with comparatively lower significance scores. Our approach is based on the attentive graph pooling layer proposed in [177] for the node classification task. We apply the graph pooling layer to the output of the GAT layer, as illustrated in Figure 6.2. An attentive graph pooling layer is included in all three GAT blocks to generate more discriminative graphs by selecting a subset of the most informative nodes and by dropping less irrelevant nodes to reduce the computation power. An example of graph pooling is illustrated in Figure 6.4.

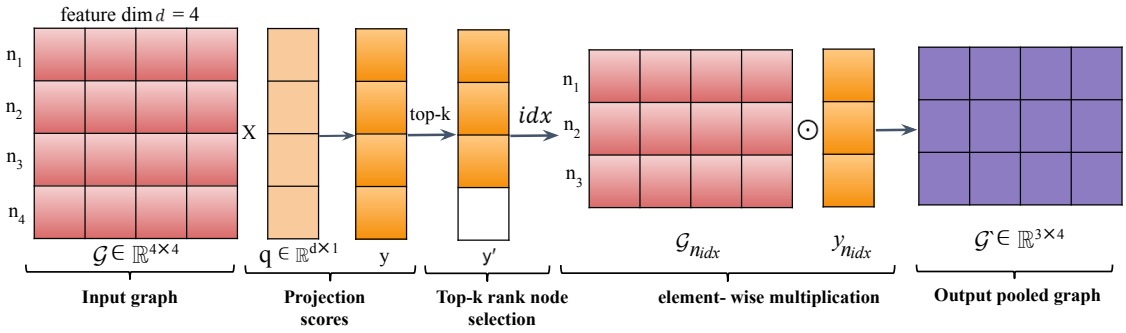


Figure 6.4: An illustration of the graph pooling layer [177]. The input graph $\mathcal{G} \in \mathbb{R}^{N\times d}$ has $N = 4$ nodes and feature dimensionality $d = 4$. $q \in \mathbb{R}^{d\times1}$ is a learnable projection vector (attention weight vector). Projection scores y are obtained from the dot product between q and the node feature vector in input graph $\mathcal{G}$ for each node n. The indices idx corresponding to the nodes with the top-k highest projection scores are used to form a new pooled graph $\mathcal{G}'$ from the element-wise multiplication of $\mathcal{G}_{n_{idx}}$ with sigmoid($y_{n_{idx}}$) (Eq. 6.8).

Let $\mathcal{G}, \mathcal{G} \in \mathbb{R}^{N\times d}$ be the input graph for a graph attention layer where N is the number of nodes and where d refers to the feature dimensionality of each node. Note that the order of nodes in the graph is meaningless; the relationships between them are defined via the attention weights assigned to each edge. Attention weights are derived via $\mathcal{G} \cdot q$, where $[\cdot]$ is the dot product and $q \in \mathbb{R}^d$ is a projection vector that returns a scalar attention weight for each node. Graph pooling uses an attention weight vector $q \in \mathbb{R}^{d\times1}$. The dot-product between the input graph $\mathcal{G}$, and q gives learnable projection scores y:

$$y = \mathcal{G} \cdot q \tag{6.7}$$

Nodes in the input graph $\mathcal{G}$ corresponding to the top-k-hot vectors y' are then retained according to element-wise multiplication:

$$\mathcal{G}'_{\text{pooled}} = \mathcal{G}_{n_{idx}} \odot \text{sigmoid}(y_{n_{idx}}), \tag{6.8}$$

where the pooling ratio k is a hyperparameter, and where $\mathcal{G}_{n_{idx}}$ and $y_{n_{idx}}$ are the node features and projection scores selected corresponding to the highest top-k indices, idx. After the multiplication of a sigmoid non-linearity with the corresponding k nodes, the nodes with the top-k values are retained while the rest are dropped.

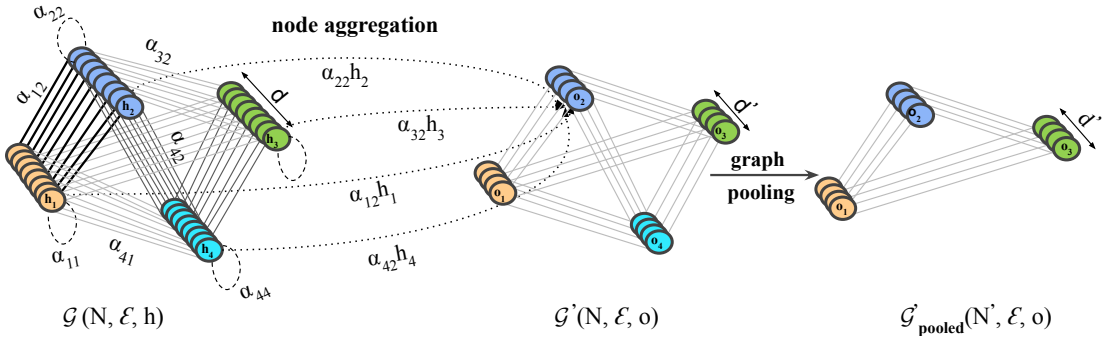


Figure 6.5: An illustration of the proposed graph-based process including a GAT and a graph pooling layer for an input graph with $N = 4$ nodes, each of node dimension $d = 8$. Circles with different colours denote the d-dimension node features h_n. A self-attention mechanism is applied on the node features to learn the influence among different nodes and to estimate attention weights α between each node pair. The edges between the neighboring node pair indicate that a learned pairwise relationship is used to calculate the relative importance between nodes using self-attention mechanism. Thicker lines (see in left-most graph) indicate higher attention weights for the given node pairs. Then, the information from the node itself and neighboring nodes are aggregated, and projected to an output feature space $\mathrm{o}_n \in \mathbb{R}^{d'}$ with target dimensionality $d' = 4$. The graph pooling layer reduces the number of nodes to improve discrimination. (Best viewed in colour.)

The graph pooling operator is defined as a function Gpool which maps a graph $\mathcal{G}$ to a new pooled graph $\mathcal{G}'_{\text{pooled}} \in (N', \mathcal{E}', \mathbf{o})$:

$$\mathcal{G}'_{\text{pooled}} = \text{Gpool}(\mathcal{G}) \tag{6.9}$$

Finally, pooled graph representations $\mathcal{G}'_{\mathbf{f}_{\text{pooled}}}$ and $\mathcal{G}'_{\mathbf{t}_{\text{pooled}}}$ are generated from the original spectral $\mathcal{G}'_{\mathbf{f}}$ and temporal $\mathcal{G}'_{\mathbf{t}}$ output graphs (described in Section 6.4.2), respectively. Since spectral $\mathcal{G}'_{\mathbf{f}_{\text{pooled}}}$ (N'_{f}=14) and temporal $\mathcal{G}'_{\mathbf{t}_{\text{pooled}}}$ (N'_{t}=23) pooled graphs have different numbers of nodes, both sets of graph nodes are projected into the same dimensional space using an affine-transform. This is accomplished using two fully-connected layers, each of which projects the node of the graph to the same node dimensionality. Figure 6.5 illustrates an example of a graph learning process, where a GAT layer operating upon an input graph $\mathcal{G}(N, \mathcal{E}, \mathrm{h})$ with N= 4 nodes. The GAT layer aggregates node information using self-attention weights between neighboring node pairs to produce an output graph $\mathcal{G}'(N, \mathcal{E}, \mathrm{o})$. Subsequently, a graph pooling layer is applied to the output graph to retain the most discriminative nodes, resulting in N'= 3 nodes as illustrated in Figure 6.5.

6.4.4 Model-level combination

A spectral and temporal graph combination module represents the forth contribution of this work. Inspired by the idea of multi-model fusion used in emotion classification [179], we simply combine two different graphs at the model-level. Model-level graph combinations (ellipse in Figure 6.2) are used to exploit complementary information captured by the spectral and temporal attention graphs. We studied three different traditional approaches to graph combination:

$$\mathcal{G}_{\mathbf{ft}} = \begin{cases} \mathcal{G}'_{\mathbf{f}_{\text{pooled}}} \oplus \mathcal{G}'_{\mathbf{t}_{\text{pooled}}} \\ \mathcal{G}'_{\mathbf{f}_{\text{pooled}}} \odot \mathcal{G}'_{\mathbf{t}_{\text{pooled}}} \\ \mathcal{G}'_{\mathbf{f}_{\text{pooled}}} \;||\; \mathcal{G}'_{\mathbf{t}_{\text{pooled}}} \end{cases} \tag{6.10}$$

where the combined graph $\mathcal{G}_{\mathbf{ft}} \in \mathbb{R}^{N_{\mathbf{ft}} \times d_{\mathbf{ft}}}$ is generated from one of the fusion approaches in Eq. 6.10. It acts to combine the spectral pooled graph $\mathcal{G}'_{\mathbf{f}_{\text{pooled}}}$ with the temporal pooled graph $\mathcal{G}'_{\mathbf{t}_{\text{pooled}}}$. A combined graph $\mathcal{G}_{\mathbf{ft}}$ contains a set of $N_{\mathbf{ft}} = 12$ nodes and each have feature dimensionality $d_{\mathbf{ft}} = 32$. The three different operators in Eq. 6.10 are element-wise addition $\oplus$, multiplication $\odot$ and concatenation $||$. A third GAT layer is then applied to $\mathcal{G}_{\mathbf{ft}}$ to produce output graph $\mathcal{G}'_{\mathbf{ft}} \in \mathbb{R}^{N_{\mathbf{ft}} \times d'_{\mathbf{ft}}}$, where $d'_{\mathbf{ft}} = 16$ is the output feature dimensionality. Graph pooling is then applied one last time to generate a pooled graph $\mathcal{G}'_{\mathbf{ft}_{\text{pooled}}}$. The final two-class prediction (bona fide or spoofed) is then obtained using linear projection and output layers. A summary of the RawGAT-ST model illustrated in Figure 6.2 is presented in the Table 6.1.

6.5 Experiments

All experiments were conducted using the ASVspoof 2019 LA database [49, 53] described in Section 2.1. The baseline is the end-to-end RawNet2 system [176] described in Chapter 5. In contrast to the baseline RawNet2 system and in order to reduce computational complexity, we reduced the number of filters to 70 in the sinc-layer for RawGAT-ST model. Like the baseline RawNet2 system, it also operates directly upon raw waveform inputs. To improve generalisation, we added channel masking in similar fashion to frequency masking [87, 107, 180] to mask (set to zero) the output of a random selection of contiguous sinc channels during training. The same channel mask is applied to all training data within each mini-batch. The number of masked channels is chosen from a uniform distribution between 0 and F_{mask}, where $F_{\text{mask}} = 14$ is the maximum number of masked channels selected based on minimum validation loss (on the ASVspoof 2019 LA development set). In contrast to usual practice, we also use fewer filters (32 and 64) in the first and second residual blocks to further protect generalisation to previously unseen attacks [181]. Graphs with a larger number of nodes will increase computation cost. To avoid this, we applied attentive graph pooling to select only the top-k

Table 6.1: The details of RawGAT-ST model architecture. Numbers denoted in sinc layer & Conv layer refer to (filter size, stride, and number of filters). The output size refers to (CNN channels, Freq, Time). Separate GAT layers are use for spectral and temporal attention blocks.

Layer	**Input**: 64600 samples	**Output shape**
Sinc-layer	Conv-1D(129,1,70)	(70,64472)
	add channel (TF representation)	(1,70,64472)
	Maxpool-2D(3)	(1,23,21490)
	BN & SeLU	
Residual-block	{ Conv-2D((2,3),1,32); BN & SeLU; Conv-2D((2,3),1,32); Maxpool-2D((1,3)) } ×2	(32,23,2387)
Residual-block	{ Conv-2D((2,3),1,64); BN & SeLU; Conv-2D((2,3),1,64); Maxpool-2D((1,3)) } ×4	S=(64,23,29)
Spectral-attention		Temporal-attention
$\max_{\mathrm{T}}(\mathrm{abs}(S)) = (64, 23)$		$\max_{\mathrm{F}}(\mathrm{abs}(S)) = (64, 29)$
GAT layer =(32,23)		GAT layer =(32,29)
Graph pooling=(32,14)		Graph pooling=(32,23)
Projection=(32,12)		Projection=(32,12)
Model-level graph combinations	element-wise addition ($\oplus$)	(32,12)
	element-wise multiplication ($\odot$)	(32,12)
	concatenation (along feature dim) ($\|\|$)	(64,12)
Spectro-temporal attention	GAT layer	(16,12)
	Graph pooling	(16,7)
	Projection (along feature dim)	(1,7)
Output	FC(2)	2

highest scoring nodes where k are empirically selected pooling ratios of 0.64, 0.81, and 0.64 for spectral, temporal and spectro-temporal attention blocks respectively.

The full model is trained using the ASVspoof 2019 LA training partition to minimise a weighted cross entropy (WCE) loss function, where the ratio of weights

Table 6.2: Results for the ASVspoof 2019 LA database are shown in terms of Pooled min t-DCF and pooled EER. Results are shown for the RawNet2 baseline system and the three variants of the RawGAT-ST system introduced in this paper.

System	Pooled min t-DCF	Pooled EER (%)
baseline	0.1300	5.64
RawGAT-ST-add	0.0373	1.15
RawGAT-ST-concat	0.0388	1.23
RawGAT-ST-mul	**0.0335**	**1.06**

assigned to bonafide and spoofed trials are 9:1 to manage the data imbalance in the training set. We used the standard Adam optimiser [155] with a mini-batch size of 10 and a fixed learning rate of 10^{-4}. The model is trained for 300 epochs. The feature extractor and back-end classifier are jointly-optimised using back-propagation [156] during training. The best model was selected based on the minimum validation loss. The proposed spectro-temporal GAT model has 0.22 M parameters and is comparatively lightweight compared to the baseline as well as other state-of-the-art systems. The code and model checkpoints are available in an open-source Python implementation.[1]

6.6 Results

Results are illustrated in Table 6.2, where columns P1 and P2 indicate results in terms of pooled min t-DCF and pooled EER for the baseline system (RawNet2) and three different variants of the proposed RawGAT-ST system. The variants involve the use of different spectro-temporal graph combination strategies. Whereas all RawGAT-ST systems outperform the baseline by a substantial margin, the best result is obtained using the RawGAT-ST-mul system for which the t-DCF is 0.0335 (cf. 0.1547 for the baseline) and the EER is 1.06% (5.54%). These results show that all RawGAT-ST systems are effective in exploiting spectro-temporal attention and model successfully and beneficially the relationships between different spectro-temporal estimates, thereby improving the discrimination between spoofed and bona fide inputs. Figure 6.6 shows that the proposed system consistently outperforms the baseline for a broad range of spoofing attacks, which is demonstrated by lower min t-DCFs and lower EERs compared to the baseline system. The RawGAT-ST-mul system gives an 87 % relative reduction in min t-DCF performance for attack A18 (which is difficult to detect with baseline RawNet2).

[1] https://github.com/eurecom-asp/RawGAT-ST-antispoofing

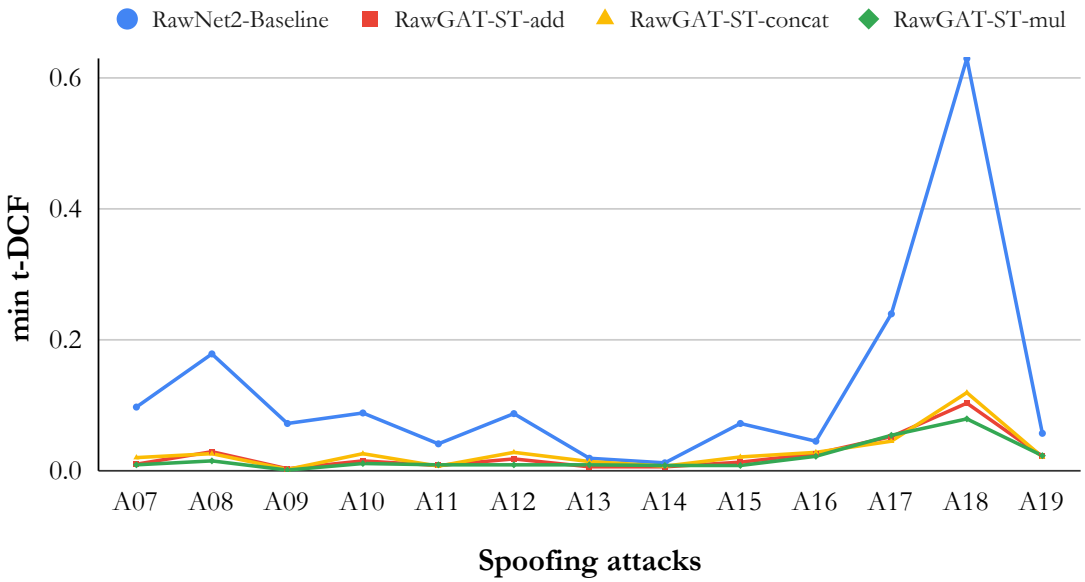


Figure 6.6: Attack-wise performance in-terms of min t-DCF for different RawGAT-ST systems along with baseline on the ASVspoof 2019 LA evaluation set.

6.7 Ablation study

Only through ablation experiments can we properly demonstrate the merit of the RawGAT-ST approach. To investigate how much the proposed spectro-temporal attention module contributes to model performance, we ran a further set of experiments while removing one of the blocks in the full RawGAT-ST architecture illustrated in Figure 6.2. Results are illustrated in Table 6.3. The top row highlighted in boldface is the RawGAT-ST-mul result that uses both spectro-temporal attention together. Ablation of the spectral GAT attention block (top left in Figure 6.2) leaves the system capable of exploiting only temporal GAT attention to

Table 6.3: Results for ablation studies

System	min-tDCF	EER
w/ spectro-temporal attention	**0.0335**	**1.06**
w/o spectral attention	0.0514	1.87
w/o temporal attention	0.0385	1.13
w/o graph pooling	0.0788	2.47

encode temporal information. Without spectral attention (third row of Table 6.3), performance degrades by 34% relative to the full system (0.0514 cf. 0.0335). The degradation in performance without temporal attention (0.0385) is less severe (13% relative), indicating the greater importance of spectral attention versus temporal attention, even though both are beneficial. Lastly, we demonstrate the benefit of graph pooling by ablating the pooling layers in all three blocks. The relative degradation in performance of 58% (0.0788 cf. 0.0335) is even more substantial and shows the benefit of using a graph pooling layer to concentrate on the most informative node features.

6.8 Performance comparison

A comparative study of RawGAT-ST to competing single systems was furthermore conducted to gauge the benefit of spectro-temporal attention in improving performance. The comparison in Table 6.4 shows that our system which uses GATs with self-attention outperforms alternative attention approaches such as a Convolutional Block Attention Module (CBAM), a traditional Squeeze-and-Excitation (SE) attention module, and a Dual attention module with pooling and convolution operations. Furthermore, to the best of our knowledge, at the time of publication our approach is the best single system reported in the literature. With only 0.22M parameters, the proposed RawGAT-ST system is among the least complex CM systems. Despite the simplicity, the proposed RawGAT-ST system outperforms all single state-of-the-art systems presented in Table 6.4. This result also points toward the benefit of operating directly upon the raw waveform and of learning the relationship between artefacts spanning spectral and temporal domains.

6.9 Summary

This chapter introduces a fully end-to-end, joint spectro-temporal graph attention network, called RawGAT-ST. This network operates directly upon raw waveform and utilises a self-attention mechanism to learn the correlation between different spectro-temporal estimates and the most discriminative nodes within the resulting graph representation. Results for the ASVspoof 2019 LA database show that the RawGAT-ST model generalises well to a diverse range of spoofing attacks, including previously unseen attacks. Our approach RawGAT-ST achieved the lowest EER at the time of publication.

Table 6.4: A comparison to recently proposed top-performing, competing state-of-the-art systems. Results reported in terms of pooled min t-DCF and EER (%).

Ref.	Front-end	Back-end	min t-DCF	EER (%)
Proposed	**Waveform**	**RawGAT-ST**	**0.0335**	**1.06**
[182]	Waveform	Res-TSSDNet	0.0481	1.64
[183]	Spec.	CNN	0.0510	1.87
[184]	Waveform	Raw PC-DARTS	0.0517	1.77
[185]	CQT	MCG-Res2Net50	0.0520	1.78
[87]	LFB	ResNet18-LMCL-FM	0.0520	1.81
[186]	LFCC	LCNN-LSTM-sum	0.0524	1.92
[88]	LFCC	Capsule network	0.0538	1.97
[187]	LFCC	Resnet18-OCsoftmax	0.0590	2.19
[89]	CQT	SE-Res2Net50	0.0743	2.50
[188]	LFCC	LCNN-Dual att.	0.0777	2.76
[189]	Waveform	RW-ResNet	0.0820	2.98
[173]	LFB	GAT-T	0.0894	4.71
[190]	LFCC	PC-DARTS	0.0914	4.96
[191]	LFCC	Siamese	0.0930	3.79
[188]	LFCC	LCNN-4CBAM	0.0939	3.67
[104]	DASC	LCNN	0.0940	3.13
[83]	LFCC	LCNN	0.1000	5.06
[192]	CQT	LCNN	0.1020	4.07
[90]	CQT	ResNet	0.1190	3.72

Chapter 7

An Integrated Spectro-temporal Graph Attention Network

In Chapter 6, we introduced the RawGAT-ST model, which assumes that the spectral and temporal graphs are homogeneous and combine them using traditional addition and multiplication operation. This model might be sub-optimal since it cannot fully exploit the different types of node information. To address this, we consider both spectral and temporal graph representations as heterogeneous graphs composed of multiple types of nodes/edges. This approach can be beneficial as it allows the model to take into account both local and global feature interactions, which may be important for accurately detecting spoofing artefacts. Furthermore, the use of a heterogeneous graph attention network enables the integration of different types of node feature representations, such as spectral and temporal characteristics which cannot be well exploited via traditional homogeneous graph combinations. To further enhance performance, we propose an extension to the RawGAT-ST model by introducing a heterogeneous stacking graph attention layer [48] leading to a new, integrated spectro-temporal graph attention network, named AASIST.

The key contributions of this work include:

- introduce an extended variant of the graph attention layer, namely a heterogeneous stacking graph attention layer (HS-GAL);
- introduce a new mechanism referred to as max graph operation (MGO);
- an exploration of a new readout scheme for graph aggregation that utilises the stack node.

7.1 Methodology

AASIST is an integrated spectro-temporal graph attention network as illustrated in Figure 7.1. It builds upon the RawGAT-ST model, whereby two heterogeneous graphs such as spectral and temporal graphs are combined at the model level. However, instead of using traditional element-wise operations as used in RawGAT-ST, AASIST employs a joint attention learner approach using the proposed heterogeneous stacking graph attention layer (HS-GAL), in addition to the proposed MGO and a new readout scheme. It also operates directly upon raw waveform inputs. First, a RawNet2-based encoder (see Section 6.4.1) is used to extract high-level feature representations $\mathrm{S} \in \mathbb{R}^{C \times F \times T}$ where C, F and T refer to the number of channels, spectral bins and temporal segments, respectively. Then, a pair of graph modules which comprise graph attention layer (GAT) and graph pooling layers is used to model the temporal and spectral features in parallel, by constructing $\mathcal{G}_{\mathrm{t}}$ and $\mathcal{G}_{\mathrm{f}}$. Both $\mathcal{G}_{\mathrm{f}}$ and $\mathcal{G}_{\mathrm{t}}$ graphs combined into a joint spectro-temporal graph $\mathcal{G}_{\mathrm{ft}}$ using a HS-GAL layer. Graph $\mathcal{G}_{\mathrm{ft}}$ is then processed by the max graph operation (MGO) which comprises four HS-GAL layers and graph pooling layers. The readout operation is performed at the node level, followed by an output layer with two nodes to distinguish bona fide from spoofed utterances. Each block in Figure 7.1 is described in the following.

7.1.1 Graph combination

Separate spectral and temporal feature representations are learned from the high-level feature representation S using a max-pooling operation which is applied to the absolute values across either temporal or spectral dimensions in order to construct either a spectral graph ($\mathcal{G}_{\mathrm{f}} \in \mathbb{R}^{N_{\mathrm{f}} \times d_{\mathrm{f}}}$) or a temporal graph ($\mathcal{G}_{\mathrm{t}} \in \mathbb{R}^{N_{\mathrm{t}} \times d_{\mathrm{t}}}$). N_{f} and N_{t} are the set of nodes, and d_{f} and d_{t} are the node dimensionalities for spectral and temporal graphs respectively. Spectral $\mathcal{G}_{\mathrm{f}}$ and temporal $\mathcal{G}_{\mathrm{t}}$ graphs are derived according to:

$$\mathcal{G}_{\mathrm{t}} = \mathrm{graph\_module}(\max_{\mathrm{F}}(\mathrm{abs}(\mathrm{S}))) \tag{7.1}$$

$$\mathcal{G}_{\mathrm{f}} = \mathrm{graph\_module}(\max_{\mathrm{T}}(\mathrm{abs}(\mathrm{S}))) \tag{7.2}$$

where each graph_module (grey boxes in Figure 7.1) comprise a GAT and a graph pooling layers. We generate a combined graph ($\mathcal{G}_{\mathrm{ft}}$) by adding edges between every node in $\mathcal{G}_{\mathrm{f}}$ and $\mathcal{G}_{\mathrm{t}}$, vice versa. This results in a heterogeneous graph ($\mathcal{G}_{\mathrm{ft}}$) with N_{f}+N_{t} nodes. The new edges in the combined graph ($\mathcal{G}_{\mathrm{ft}}$) allow for the estimation of attention weights between pairs of heterogeneous nodes which each span both spectral and temporal domains. Graph combination enables the concurrent modeling of heterogeneous graph representations with different node dimensions.

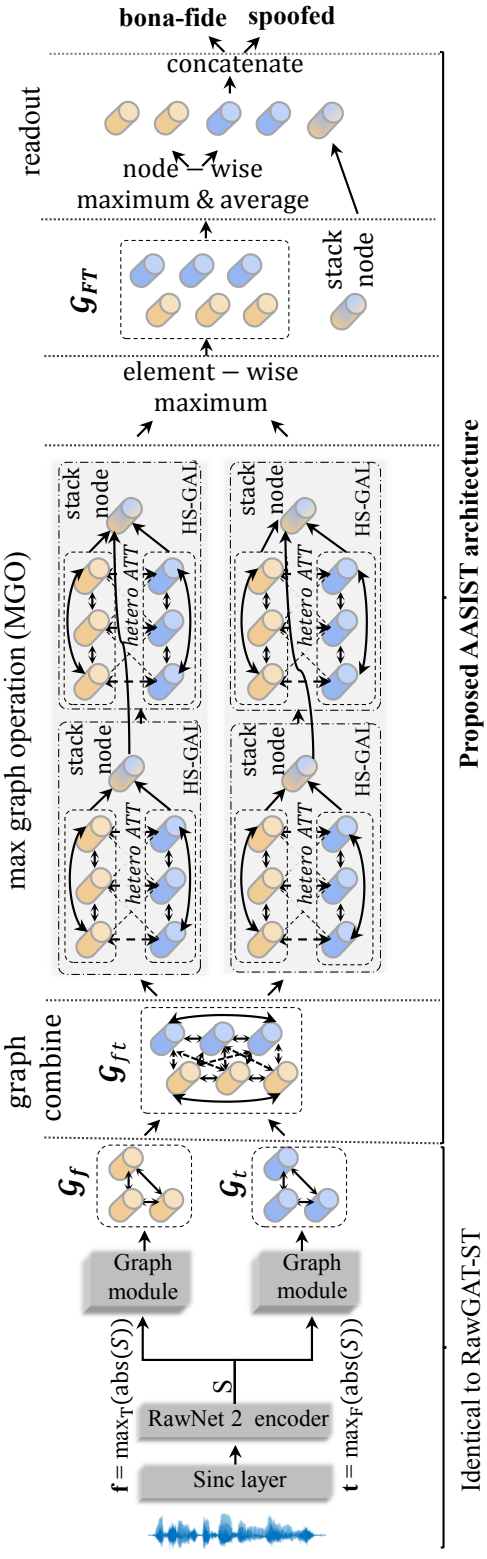


Figure 7.1: Depiction of how AASIST model generates the two class predictions output. Here, Spectral (f) and temporal (t) feature representations from RawNet2 encoder; two parallel graph modules are used to model spectral and temporal graphs. Proposed approach: we generate a combined heterogeneous graph $\mathcal{G}_{\text{ft}}$ using two graph module outputs; a MGO technique is applied to two branches that model heterogeneous graphs in parallel before the application of an element-wise maximum; each branch includes two HS-GAL layers and two graph pooling layers (graph pooling layers and one HS-GAL layer is omitted in the illustration); the readout scheme perform node-wise maximum and average, and concatenate them with the stack node.

7.1.2 Heterogeneous stacking graph attention layer

An HS-GAL layer contains an attention mechanism modified in order to accommodate graph heterogeneity [48] and an additional stack node [193]. Our use of a heterogeneous attention layer is inspired by the approach to modeling heterogeneous data described in [48]. First, $\mathcal{G}_{\mathrm{t}}$ and $\mathcal{G}_{\mathrm{f}}$ are projected using an affine-transformation to another latent space with common dimensionality d_{ft} before being fed as input to the HS-GAL layer. Unlike RawGAT-ST, which used a single projection vector to derive attention weights between node pairs, AASIST uses three different projection vectors to estimate the attention weights between heterogeneous graphs. These projection vectors are used to learn attention weights for edges connecting: (i) nodes in $\mathcal{G}_{\mathrm{f}}$ to other nodes in $\mathcal{G}_{\mathrm{f}}$ (intra-connection between orange nodes); (ii) nodes in G_{f} to nodes in G_{t} and nodes in G_{t} to nodes in G_{f} (inter-connection between orange and blue nodes (dotted edges)); (iii) nodes in $\mathcal{G}_{\mathrm{t}}$ to other nodes in $\mathcal{G}_{\mathrm{t}}$ (intra-connection between blue nodes). A heterogeneous graph ($\mathcal{G}_{\mathrm{ft}}$) comprising two types of nodes, temporal nodes (blue nodes in Figure 7.1) and spectral nodes (orange nodes), with different node dimensionality. The stack node is used to accumulate heterogeneous information between spectral and temporal graphs. The stack node is connected to all other nodes (stemming from G_{f} and G_{t}), and we use uni-directional edges from all other nodes to the stack node to preserve information in both G_{f} and G_{t}. Stack node does not transmit information to other nodes.

7.1.3 Max graph operation

The max graph operation (MGO) layer, illustrated in Figure 7.1, is inspired by the popularity in the anti-spoofing literature [194] of element-wise maximum operations using max feature map (MFM) [80]. The motivation behind MGO is to enable different branches to learn different groups of artefacts in parallel. In our proposed framework, HS-GALs are applied with a MGO layer consisting of two branches, each consisting of two HS-GALs in sequence. A graph pooling layer is applied to the output of each HS-GAL layer. Thus, the MGO layer comprises a total of four HS-GALs and four graph pooling layers. An element-wise maximum operation is applied to the outputs of each branch to produce another heterogeneous graph $\mathcal{G}_{\mathrm{FT}}$. HS-GALs in each branch share a common stack node. The stack node of each preceding HS-GAL is fed to the following HS-GAL so that information in both temporal and spectral graphs is preserved.

7.1.4 Readout scheme

The readout scheme (penultimate block in Figure 7.1) performs node-wise maximum and average operations on the heterogeneous graph $\mathcal{G}_{\mathrm{FT}}$. The output of the

Table 7.1: The AASIST model architecture and configuration. Output dimensions refer to (channels, frequency, time). Batch normalisation (BN) and scaled exponential linear unit (SeLU).

	Layer	Input:64600 samples		Output shape	
RawNet2 encoder	Sinc-layer	Conv-1D(129,1,70)		(70,64472)	**Identical to RawGAT**
		add channel (T-F representation)		(1,70,64472)	
		Maxpool-2D(3)		(1,23,21490)	
		BN & SeLU			
	Residual-block	Conv-2D((2,3),1,32) BN & SeLU Conv-2D((2,3),1,32) Maxpool-2D((1,3))	×2	(32,23,2387)	
	Residual-block	Conv-2D((2,3),1,64) BN & SeLU Conv-2D((2,3),1,64) Maxpool-2D((1,3))	×4	S=(64,23,29)	
	Spectral-attention			Temporal-attention	
	$\max_{\mathrm{F}}(\mathrm{abs}(S))$ =(64,23)			$\max_{\mathrm{T}}(\mathrm{abs}(S))$ =(64,29)	
	$\mathcal{G}_f$ = (64(d_f),11(N_{f}))			$\mathcal{G}_t$ = (64(d_t),20(N_{t}))	
	Hetero. graph ($\mathcal{G}_{ft}$)	HS-GAL		(64(d_{ft}),31(N_{ft}))	**AASIST**
	HS-GAL→HS-GAL, stack node (32(d_{ft}), 15(N_{ft})), (32,)			HS-GAL→HS-GAL, stack node (32(d_{ft}), 15(N_{ft})), (32,)	
	MGO ($\mathcal{G}_{FT}$)	element-wise max.		(32(d_{ft}), 15(N_{ft})), (32,)	
	readout	node-wise max. and avg. & concatenation		(160(d_{ft}),)	
	Output	FC(2)		2	

readout layer is formed from the concatenation of five nodes. The first four nodes are derived by applying a maximum and average operations to the spectral nodes (orange) and temporal nodes (blue) of $\mathcal{G}_{\mathrm{FT}}$. The last is the copied stack node. The fully connected layer with two nodes is used to generate a two-class prediction output (bona fide and spoofed). A summary of the AASIST model illustrated in Figure 7.1 is presented in Table 7.1.

7.2 Experiments

Experiments were performed using the ASVspoof 2019 LA dataset [49] described in Section 2.1. We used RawGAT-ST described in Chapter 6 as a baseline. The performance of spoofing detection systems can vary significantly with different random seeds as reported in [91]. In order to examine the variability in model performance, we trained and evaluated each model independently three times with different random seeds. Audio utterances are cropped or concatenated to give segments of $\approx$ 4 seconds duration (64,600 samples) which are then fed to the RawNet2-based encoder to extract high-level representations. The encoder comprises a first layer of time-domain sinc-convolution [144] with 70 filters, and a series of six residual blocks. As illustrated in Table 7.1, the first two residual blocks contain 32 filters, whereas the remaining four contain 64 filters. The first two GAT layers have node dimensions of 64 (d_{f} and d_{t}) in graph modules, whereas the following GAT layers have node dimensions of 32 (d_{ft}). Graph pooling layers were applied to remove 30% and 50% of temporal and spectral nodes, respectively. Subsequent GAT layers are followed by graph pooling layers which further reduce the number of nodes by 50%. Adam optimiser [195] was used with a learning rate of 10^{-4}. The model was trained for 100 epochs.

7.3 Results

Results are illustrated in Table 7.2, where the last two columns (P1 and P2) indicate the pooled min t-DCF and pooled EER results for the RawGAT-ST baseline and AASIST model, respectively. Results show the average of three runs of each experiment with different random seeds. The single best performance is provided in parentheses. They show that the proposed AASIST model significantly outperforms the RawGAT-ST baseline system by substantial margins. The best result was obtained using the AASIST model for which t-DCF is 0.0275 (0.0335 for the baseline) and the EER is 0.83% (1.19% for the baseline). In the best case, AASIST improves upon the baseline by over 20% relative in terms of pooled min t-DCF.

Table 7.2: Breakdown results in terms of EER (%) for all 13 attacks in the ASVspoof 2019 LA evaluation set, pooled min t-DCF (P1), and pooled EER (%, P2) for the baseline RawGAT-ST and the proposed AASIST models. Results are the average (best) obtained from three runs of each experiment with different random seeds.

System	**A07**	**A08**	**A09**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**P1**	**P2**
RawGAT	0.34	0.52	0.08	0.39	0.28	0.34	0.30	0.28	0.26	0.81	1.74	3.08	0.70	0.040 (0.033)	1.28 (1.06)
AASIST	0.80	0.44	0.00	1.06	0.31	0.91	0.1	0.14	0.65	0.72	1.52	3.40	0.62	0.035 (**0.027**)	1.13 (**0.83**)

Table 7.3: Results for ablation studies. Performance reported in terms of pooled min t-DCF and EER and for "average(best)" results from three experiments with different random seeds.

Configuration	min t-DCF	EER
AASIST	**0.0347 (0.0275)**	**1.13 (0.83)**
w/o heterogeneous attention	0.0415 (0.0384)	1.44 (1.37)
w/o stack node (conventional readout)	0.0380 (0.0330)	1.21 (1.03)
w/o MGO	0.0410 (0.0378)	1.35 (1.19)

7.4 Ablation study

Table 7.3 presents the results of ablation experiments, in which we remove individual blocks or operations from the full AASIST architecture to evaluate their contribution to performance. Results show that all three proposed techniques, the heterogeneous stacking graph attention layer, MGO and stack node are beneficial to spoof detection. By removing any of them, results are worse than for the full AASIST architecture. The heterogeneous attention layer has the most significant impact on performance. The impact of MGO is also substantial, with a relative degradation of 27% (0.0378 cf. 0.0275) when removed from the AASIST model. This demonstrates the benefit of MGO in allowing different branches to learn different groups of artefacts. The stack node also contributes positively to performance.

7.5 Performance comparison

Table 7.4 presents a comparison of the proposed AASIST model to the performance of the top-4 competing single CM systems from the literature [182, 184, 186]. The set of systems covers a broad range of different front-end representations and different model architectures. As shown in Table 7.4, four of the top five systems operate upon raw waveform inputs while the top two systems are based upon graph attention networks. The proposed AASIST system is also among the least complex system. To the best of our knowledge, and at the time of writing, AASIST was the best single CM system reported in the literature.

7.6 Summary

This chapter introduces an extension to the RawGAT-ST model by introducing a heterogeneous graph attention layer leading to a new, integrated spectro-temporal graph attention network, called AASIST. The model incorporates a heterogeneous stacking graph attention layer which allows for the modeling of a relationship be-

Table 7.4: A comparison to recently proposed, competing state-of-the-art systems. Results reported in terms of pooled min t-DCF and pooled EER (%). For the proposed AASIST model, we report the best single result. All systems shown are single models without any kind of score-level fusion.

Ref.	Front-end	Model	Min t-DCF	EER (%)
Ours	**Waveform**	**AASIST**	**0.027**	**0.83**
Ours	Waveform	RawGAT-ST	0.033	1.06
[196]	CQT	Non-OFD	-	1.35
[182]	Waveform	Res-TSSDNet	0.048	1.64
[184]	Waveform	Raw PC-DARTS	0.052	1.77

tween different nodes and edges in heterogeneous spectral and temporal graph representation using a self-attention mechanism. The proposed approach demonstrates 20% relative improvement over the baseline in terms of min t-DCF. AASIST is one of the least complex models among all the CM solutions reported in the literature to date.

Chapter 8

Data Augmentation

With more powerful TTS and VC techniques emerging, the robustness and generalisation of spoofing countermeasures has become a critical challenge. Generalised spoofing detection solutions are especially important for real-world scenarios in which one can expect spoofing attacks. The results presented earlier in this thesis, together with other work [41, 197, 198] has shown that the fundamental differences between training and testing data lead to a substantial difference in performance, indicating a persisting lack of generalisation in the face of previously unseen spoofing attacks. To reduce this performance gap and improve generalisation, we propose a novel data augmentation technique to produce additional data to train countermeasures. This approach can help to reduce over-fitting and improve reliability and domain robustness, particularly in the face of previously unseen spoofing attacks.

8.1 Motivation

The recent ASVspoof 2021 challenge [199] focused on the problem of spoofing detection in a challenging logical access scenario, where both bona fide and spoofed utterances are encoded and transmitted across telephony (PSTN+VoIP) networks. The goal was to develop reliable detection solutions using only the training and development partitions of the ASVspoof 2019 LA dataset, neither of which includes encoding or transmission effects. There is hence a need for data augmentation techniques to compensate for the lack of in-domain training and development data. The work presented in this chapter focuses on our solution to address transmission, and channel variability issues by introducing a novel raw data augmentation technique called *RawBoost*. The goal of this work is to improve spoofing detection reliability in the face of nuisance variation stemming from unknown encoding, and transmission conditions and from different microphones and amplifiers, and both linear and non-linear device-generated distortion, all of

which characterise a logical access or telephony scenario.

The key contributions of this work include:

- Propose a novel raw data-augmentation technique to introduce the variability in the training dataset;
- Explore simple linear and non-linear signal processing algorithms to incorporate different noise variations in the training data online.

8.2 Related work

Data augmentation (DA) is commonly applied in many machine learning tasks to generate new samples from a source database, here utterances, to augment the pool of data available for training. The use of additional augmented data which exhibits variability not contained in the source data can help to reduce model over-fitting and bias, and hence improve classification performance. Nowadays, DA is an integral component of modern machine learning pipelines and has been applied successfully in a host of different machine learning fields, such as image processing [200], speech recognition [201, 202] and speaker verification [203]. Recent work has also demonstrated its use in anti-spoofing [87, 95, 98, 101, 103–105]. A number of approaches to DA have been proposed in the literature, e.g., random cropping, rotation and mirroring for image-related tasks [204]; speed perturbation, pitch shifting, time stretching, random frequency filtering, reverberation, text-to-speech data augmentation and vocal tract length transformations for speech-related tasks [205, 206].

Knowing that ASVspoof 2021 evaluation data would contain both bona fide and spoofed utterances treated with a variety of unknown codecs and compression effects, ASVspoof 2021 challenge participants used DA techniques such as speed perturbation [201], SpecAugment [107], codec augmentation [202] and SpecMix [207] to improve performance. SpecAugment and SpecMix DA techniques are suitable for spoofing detection models which operate on 2-D feature representations. The current trend is towards raw end-to-end techniques for spoofing detection [101, 105, 146, 176, 184, 208]. Hence, there is a need for DA techniques that account for the variability expected in logical access or telephony scenarios and, in particular, techniques that can also be applied at the raw waveform level. We hence proposed a DA technique that is compatible with our end-to-end models such as RawNet2, RawGAT-ST and AASIST presented in Chapters 5, 6 and 7 respectively.

8.3 RawBoost

RawBoost[1] is a data boosting and augmentation technique which operates directly upon raw waveforms. Data boosting can encode prior knowledge about data or task-specific invariances, act as a regulariser to prevent over-fitting, and can improve model robustness [209]. RawBoost is built upon simple signal processing techniques and is less complex with regard to the other standard DA techniques using deep neural networks. Unlike WavAugment [206], a approach to DA through band-reject filtering, and reverberation (libsox library) or additive noises (MUSAN [210] and AudioSet [211]), RawBoost operates upon the waveform without the need for any external data resources. It generate augmented data online (on-the-fly) from the existing source database. RawBoost uses established linear and non-linear signal processing techniques to boost or distort a set of utterances in a training dataset. The RawBoost framework is illustrated in Figure 8.1 and comprises the three independent DA processes. They are described in the following:

① Linear and non-linear convolutive noise

Any channel involving some form of encoding, compression, decompression and transmission introduces stationary convolutive distortion. Most such channels will also introduce non-linear disturbances which are themselves subject to a stationary convolutive distortion, but of different characteristics (see [53], Figure 6). In order to improve robustness to such nuisance variation, we explored the combination of multi-band filtering with Wiener-Hammerstein systems (one linear and one non-linear filter) [212]. Hammerstein systems are proven, popular models of non-linear systems in which non-linear static and linear dynamic subsystems are separated into different orders [212]. While Hammerstein models estimate multi-band filters from the response of non-linear systems, here we use the same idea to generate signal distortions.

Multi-band filters are designed to generate convolutive noise (CN noise) using time domain notch filtering. They are applied to a single utterance at a time and with a set of N_{notch} notch filters, each with a randomly-chosen center frequencies f_c and filter widths Δf. A single finite impulse response (FIR) filter with a randomly-chosen gain g_j^{cn} is then defined using a window-based filter design method [213], resulting in a filter with the desired frequency response using a randomly-chosen number of filter coefficients N_{fir}. The higher the number of coefficients, the more abrupt the frequency response; filters with fewer coefficients will exhibit passband ripple or distortion in addition to smoother cut-in and cut-off responses. An ex-

[1] https://github.com/TakHemlata/RawBoost-antispoofing

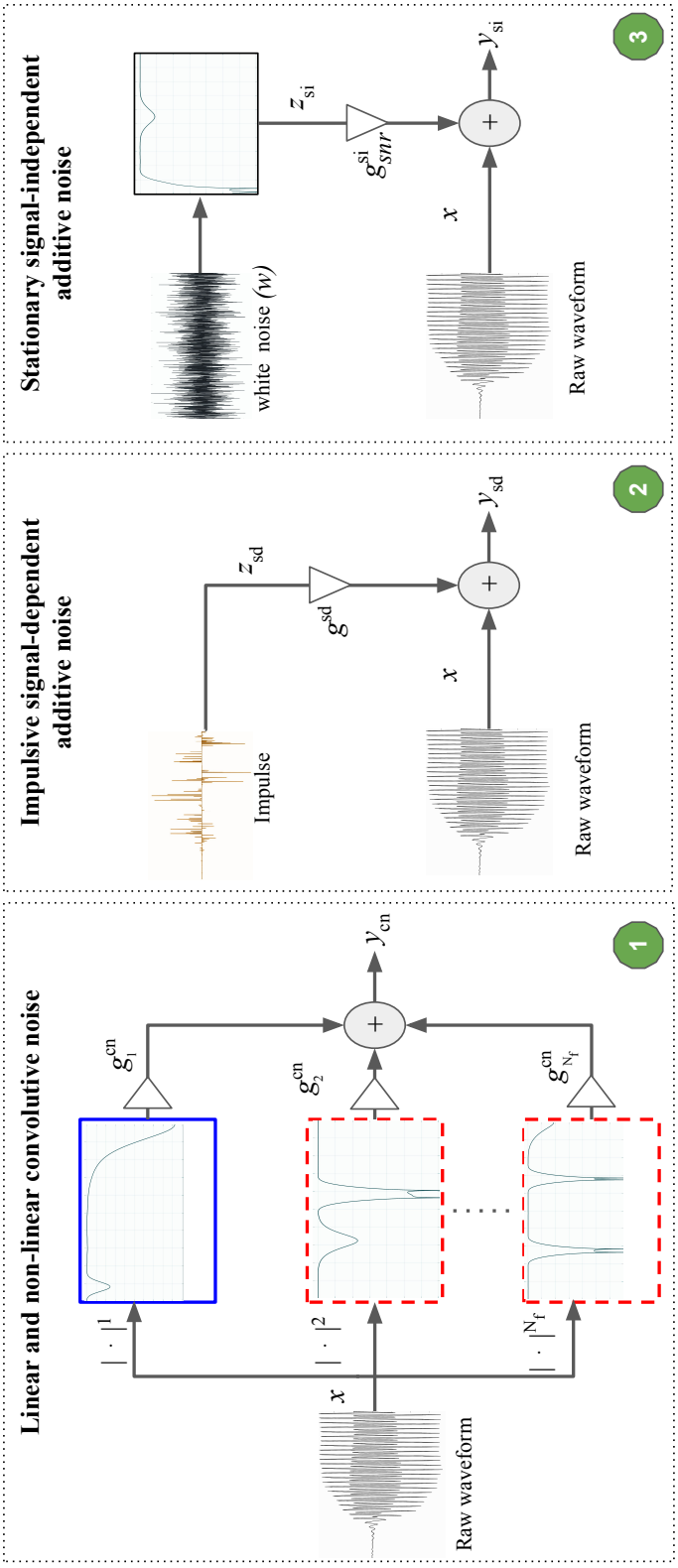


Figure 8.1: Proposed RawBoost DA framework including: ① linear and non-linear convolutive noise; ② impulsive signal-dependent additive noise; ③ stationary signal-independent additive noise. In ①, the profile in each rectangular box shows the frequency response for the first harmonic (linear, solid blue box) and higher order harmonics (non-linear, dashed red boxes).

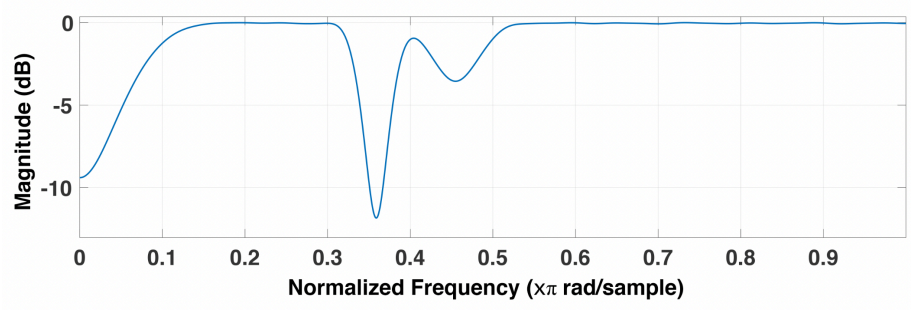


Figure 8.2: Magnitude response of a multi-band filter with $N_{\text{notch}} = 3$ notch filters centered at normalised frequencies of 0.01, 0.35 and 0.45, bandwidths 0.06, 0.03 and 0.02 and number of filter coefficients 30, 94 and 52.

ample filter frequency response is illustrated in Figure 8.2. It has $N_{\text{notch}} = 3$ notch filters, each with different center frequencies, stop-band widths and number of filter coefficients.

Hammerstein systems generate higher-order harmonics whereby a component f_0 in the input to a non-linear system is supplemented at the output by $N_{\text{f}} - 1$ new components at $2f_0, 3f_0, ..., N_{\text{f}}f_0$, leading to non-linear harmonic distortion. The frequency and amplitude of each higher-order harmonic are dependent upon those of the original component and the characteristics of the non-linear system. Convolutive noise y_{cn} (See (1) in Figure 8.1), is generated according to:

$$y_{\text{cn}}[n] = \sum_{j=1}^{N_{\text{f}}} g_j^{\text{cn}} \sum_{i=0}^{N_{\text{fir}_j}} b_{i_j} \cdot x^j[n-i], \tag{8.1}$$

where $x \in [-1, 1]^{l \times 1}$ denotes a raw waveform of l samples, $j \in [1, N_{\text{f}}]$ is the order of the (non-)linearity ($N_{\text{f}} = 1$ refers to the filter applied to the linear component x), b_{i_j} denotes the coefficients of the j^{th} multi-band filter.

(2) Impulsive signal-dependent additive noise

Impulsive signal-dependent (ISD) noise is commonly introduced through data-acquisition, resulting from, e.g., clipping, non-optimal device operation (microphones and amplifiers), synchronisation and overflow issues, or as a result of insufficient computational power. It is typically orders of magnitude lower in amplitude than signal-independent noise [214]. We model such nuisance variability as non-stationary impulsive disturbances (See (2) in Figure 8.1) consisting of instantaneous or impulse-like amplitude variations. The disturbance z_{sd} is applied to a maximum of $P \leq l$ uniformly distributed samples $\{p_1, p_2, ..., p_P\}$ in x to obtain y_{sd} according

to:

$$y_{\text{sd}}[n] = x[n] + z_{\text{sd}}[n], \tag{8.2}$$

where

$$z_{\text{sd}}[n] = \begin{cases} g^{\text{sd}} \cdot D_R\{-1,1\}[n] \cdot x[n], & \textit{if } n = \{p_1, p_2, ..., p_P\} \\ 0, & \text{otherwise} \end{cases} \tag{8.3}$$

is a *signal-dependent additive noise* component, $g^{\text{sd}} > 0$ is a signal dependent gain parameter and where $D_R\{-1,1\}[n]$ denotes P values randomly chosen from the distribution:

$$f_R(r) = \begin{cases} -log(r), & 0 < r \leq 1 \\ -log(-r), & -1 \leq r < 0 \end{cases} \tag{8.4}$$

For convenience, the maximum number of samples P is chosen relatively as $P_{\text{rel}} = P/l$.

(3) Stationary signal-independent additive noise

The use of signal-independent additive (SIA) noise is one of the most popular forms of DA approaches and has been applied in a wide variety of applications, including speech recognition [215], speaker recognition [216], emotion recognition [217], as well as audio forgery [218] and spoofing detection [104, 219, 220]. SIA noise can result from loose or poorly joined cable connections, transmission channel effects, electromagnetic interference or thermal noise. In contrast to the generation of impulsive noise, a stationary white noise w (See (3) in Figure 8.1) is coloured using a FIR filter designed in the same way as described in Section 3.1, before being added to the entire utterance:

$$y_{\text{si}}[n] = x[n] + g^{\text{si}}_{snr} \cdot z_{\text{si}}[n], \tag{8.5}$$

where

$$g^{\text{si}}_{snr} = \frac{10^{\frac{SNR}{20}}}{\|z_{\text{si}}\|^2 \cdot \|x\|^2} \tag{8.6}$$

is a gain parameter corresponding to a randomly chosen SNR and where z_{si} is the result of white noise w coloured by the FIR filter.

8.4 Experiments

Experiments were conducted using the ASVspoof 2021 LA database described in Section 2.1. As per the ASVspoof 2021 Challenge rules [199, 221], we used only the ASVspoof 2019 LA training data to train our spoofing CM and used development data for validation. The baseline is the end-to-end RawNet2[2] system described in

[2] https://github.com/asvspoof-challenge/2021/tree/main/LA/Baseline-RawNet2

Table 8.1: The RawNet2 architecture used for anti-spoofing. BN refers to batch normalisation. The output size refers to (CNN channels, Freq., Time).

Layer	**Input**: 64600 samples	**Output shape**
Sinc layer	Conv(1025,1,20) Maxpooling(3) BN & SeLU	(21192,20)
Res block 1	{ BN & LeakyReLU Conv(3,1,20) BN & LeakyReLU Conv(3,1,20) Maxpooling(3) FMS } × 2	(2354,20)
Res block 2	{ BN & LeakyReLU Conv(3,1, 128) BN & LeakyReLU Conv(3,1, 128) Maxpooling(3) FMS } × 4	(29,128)
GRU	GRU(1024)	(1024)
FC	1024	(1024)
Output	1024	2

Chapter 5. To reduce computational complexity, we reduced the number of sinc filters in the first layer to 20. We found that a larger number of coefficients in sinc filters results in better detection performance; hence, we used a filter length of 1025 (number of filter coefficients) for each sinc filter. Full architecture details are summarised in Table 8.1 and also described in Chapter 5.

RawBoost parameters are generated according to the configuration illustrated in Table 8.2 for each of the three techniques. Values expressed within ranges are drawn from the corresponding uniform distributions. Each technique is applied alone as well as in different combinations and in both series and parallel. For series combinations, the output of one technique is used as the input to the next. For parallel combinations, an original input utterance is treated independently with each technique before the resulting distortions are combined. Output waveforms are normalised to prevent overflow. In our experiments, we used RawBoost to add nuisance variability on-the-fly to *existing* training data, instead of to generate *additional* data. RawBoost parameters and ranges illustrated in Table 8.2 were then selected based on the results of experimentation involving boosted and augmented training and development data only. Other state-of-the-art DA techniques such

Table 8.2: RawBoost parameter values. Values within expressed ranges are selected at random (uniform distributions).

Param.	$N_{\mathbf{notch}}$	$N_{\mathbf{fir}}$	N_f	f_c **[Hz]**	Δf **[Hz]**	$g_1^{\mathbf{cn}}$ **[dB]**	$g_{2-N_f}^{\mathbf{cn}}$ **[dB]**	$P_{\mathbf{rel}}$ **[%]**	$g^{\mathbf{sd}}$	SNR **[dB]**
(1)	5	[10,100]	5	[20,8000]	[100,1000]	[0,0]	[-5,-20]	-	-	-
(2)	-	-	-	-	-	-	-	[0,10]	2	-
(3)	5	[10,100]	1	[20,8000]	[100,1000]	-	-	-	-	[10,40]

as SpecAugment and WavAugment are also applied to show the effectiveness of RawBoost data augmentation. For SpecAugment experiments, frequency-masking is not applied to raw waveforms, but to the sinc filterbank output instead. Frequency masking is applied during training to mask random contiguous sinc channels, whereas time-masking is applied to the temporal segments. The number of masked filter channels is chosen from a uniform distribution between 0 and 4. The time masking length is similarly chosen, between 0 and 40 randomly chosen samples. WavAugment [206] is applied through band reject filtering, time dropping and by adding additive noises from external database such as MUSAN [210] and the AudioSet [211] in exactly similar manner as described in [206].

8.5 Results

Results are illustrated in Table 8.3 for the baseline system (row 2) without the application of any DA technique, and for the same system trained using one of the three approaches: RawBoost; WavAugment; SpecAugment. In each case, results are shown for separate augmentation techniques and a selection of combinations (column 2). Columns 3-9 show results for each evaluation condition (C1-C7). Columns 10 and 11 show the pooled min t-DCF (P1) and pooled EER (P2). Results are presented in terms of the minimum tandem detection cost function (min t-DCF) [57] and equal error rate (EER). All results are derived using updated min t-DCF metric which used in ASVspoof 2021 challenge [221]. Table 8.3 shows that all RawBoost DA strategies led to better performance than the baseline for all 7 evaluation conditions. The baseline pooled min t-DCF of 0.4257 drops to 0.3527 when using linear and non-linear convolutive noise (1), to 0.3260 using ISD additive noise (2), and to 0.3372 using stationary SIA noise (3).

The best result is obtained using the RawBoost (1)+(2) system, which achieves a 27% relative reduction in the min t-DCF (0.3099) and a 45% relative reduction in the EER (5.31%). The addition of stationary SIA noise, while beneficial on its own, did not lead to any improvements when combined with other techniques.

Table 8.3: ASVspoof 2021 LA RawNet2 results in terms of the min t-DCF for each codec conditions, pooled min t-DCF (P1) and pooled EER (P2). Data-augmentation methods, namely the proposed RawBoost, WavAugment (WavAug) and SpecAugment (SpecAug) are used for performance comparison. 'none' represent the baseline result without applying any DA technique. The series and parallel operation indicated by 'S' and 'P', respectively. All experiments were performed on the same amount of audio data.

DA	Methods	C1	C2	C3	C4	C5	C6	C7	P1	P2
none	-	0.4629	0.5594	0.7886	0.4954	0.5582	0.6774	0.5727	0.4257	9.50
Proposed: RawBoost	① CN noise	0.4531	0.5077	0.6160	0.4731	0.5019	0.5819	0.5317	0.3527	7.22
	② ISD noise	0.4373	0.5015	0.5041	0.4751	0.4920	0.5385	0.5099	0.3260	6.09
	③ SI noise	0.4544	0.5094	0.5349	0.4811	0.5036	0.5289	0.4964	0.3372	7.85
	S: ①+②	**0.4449**	**0.4806**	**0.5046**	**0.4635**	**0.4616**	**0.5025**	**0.4776**	**0.3099**	**5.31**
	P: ①+②	0.4471	0.5094	0.5507	0.4724	0.5032	0.5585	0.5243	0.3261	5.57
	S: ①+③	0.4569	0.5203	0.5576	0.4765	0.5057	0.5442	0.5134	0.3361	6.27
	P: ①+③	0.4460	0.5144	0.5306	0.4692	0.4989	0.5299	0.5190	0.3349	6.77
	S: ②+③	0.4640	0.5056	0.5100	0.4910	0.5060	0.5240	0.5171	0.3329	6.58
	P: ②+③	0.4899	0.5148	0.4889	0.5035	0.5148	0.5647	0.5381	0.3590	8.44
	S: ①+②+③	0.4437	0.4910	0.4986	0.4576	0.4937	0.5037	0.4858	0.3192	5.39
	P: ①+②+③	0.4404	0.4996	0.5072	0.4803	0.4862	0.5309	0.5140	0.3288	6.40
WavAug	(1) time-drop	0.4582	0.5049	0.5133	0.4598	0.5094	0.5296	0.4739	0.3490	8.72
	(2) band-reject	0.4763	0.5417	0.5912	0.4957	0.5387	0.5628	0.5174	0.3692	8.86
	(3) add. noise	0.5508	0.6721	0.7014	0.5531	0.6649	0.6549	0.5660	0.4819	13.38
	S: (1)+(2)+(3)	0.4652	0.4897	0.5172	0.4736	0.4802	0.5163	0.4990	0.3435	7.32
SpecAug	(1) freq-mask	0.4579	0.5292	0.7171	0.4894	0.5399	0.6642	0.5335	0.4214	9.80
	(2) time-mask	0.4581	0.5049	0.5134	0.4598	0.5094	0.5295	0.4739	0.3491	8.72
	S: (1)+(2)	0.4668	0.4985	0.5032	0.4927	0.4918	0.5162	0.4822	0.3418	8.25

Table 8.4: A performance comparison with top-performing single system for ASVspoof 2021 LA challenge in terms of pooled min t-DCF and pooled EER (%). Results are presented in order according to lower pooled min t-DCF.

System	Front-end	DA approach	min t-DCF	EER
LCNN [96]	Mel STFT	RS Mixup and FIR filtering	0.2430	2.21
ResNet-L-LDE [97]	LFB	Frequency masking, codecs, RIR, MUSAN	0.2720	3.68
Ours: RawNet2	Raw waveform	RawBoost ①+②	0.3099	5.31
SE-ResNet18 [100]	LFCC	codecs	0.3129	6.62
LCNN [98]	CQT	codecs	0.3197	5.27

This could be due to the absence of ambient noise in the ASVspoof 2021 LA dataset. The effect of DA using SIA noise matches the effects of mp3/mp4 lossy compression [222] used in ASVspoof 2021 DF database generation. Hence, it may work well for the detection of deepfake audio.

8.6 Performance comparison

Illustrated in Table 8.4 is a comparison of RawBoost performance to that of competing systems reported in the literature. To focus upon the benefits of DA, the comparison is restricted to single systems.[3] The RawNet2 system with ①+② RawBoost DA gives the third best result. Among the top three systems, only RawNet2 operates directly upon raw waveform inputs. The ResNet-L-LDE system [97], which uses three different DA techniques: i) SpecAugment (frequency masking); ii) speech codecs; iii) external noisy recordings contained from the MUSAN databases [210]. In contrast, RawBoost requires no such external data sources. The top-performing LCNN system [96] uses random square (RS) Mixup [223] and FIR filtering DA. The FIR filtering DA approach aims to emulate the application of different telephony codecs and is conceptually similar to our use of FIR filtering in RawBoost. While applied at the data level, RS Mixup is accompanied with modifications at the model level (the loss function in [96]). RawBoost requires no such intervention at the model level. The same baseline (RawNet2) system listed in Table 8.5 with different DA techniques clearly shows the effectiveness of our proposed DA technique. RawBoost is competitive with all these DA approaches while not requiring the use of any additional codec implementations.

[3]While some ensemble systems outperform those considered here, they are substantially more complex and their inclusion would compound the difficulty in assessing *data augmentation*. Unlike the comparisons made in Table 8.3, differences in Table 8.4 stem from differences in DA techniques as well as the underlying models/classifiers.

Table 8.5: A baseline performance comparison with standard data-augmentation techniques. Results for first two rows copied from Table 8.3.

Augmentation	**min t-DCF**	**EER (%)**
SpecAugment	0.3418	8.25
WavAugment	0.3435	7.32
Trans. codec	0.3297	8.17
Multimedia codec	0.3168	6.36
RawBoost	**0.3099**	**5.31**

8.7 Summary

This chapter introduces a new data augmentation technique called *RawBoost* designed to expand and enhance the available training data by modifying utterances which exhibit the variability expected in telephony scenarios. Rawboost generates augmented waveforms by perturbing a set of existing utterances with linear and non-linear convolutive, impulsive, and stationary randomly coloured additive noise. Despite its simplicity, RawBoost is also data, application and model agnostic; it operates upon an existing source database without the need for any additional external data resources, nor intervention at the model level. While this chapter demonstrates its application to improve spoofing and deepfake detection performance, it might have application to other related classification tasks where similar nuisance variability is expected, e.g. automatic speaker verification or automatic speech recognition. Additionally, an independent data augmentation algorithm can be applied easily with any machine learning framework. However, the application of these processes to raw waveforms is not yet standardised, and further research is required to achieve the maturity of boosting techniques applied directly to the models and activations, allowing the neural network to learn the appropriate Rawboost parameters for the given classification task.

Chapter 9

A Self-supervised Learning Based Front-end

It is impractical or even impossible to acquire training data that is representative of spoofing attacks with near-boundless variability. Nonetheless, the performance of spoofing countermeasure (CM) systems depends on the use of sufficiently representative training data. The results of the ASVspoof 2021 challenge [96–101, 224] shows a fundamental gap between the performance for development and evaluation data, indicating a persisting lack of generalisation in the face of unseen spoofing attacks. In this chapter, we sought to investigate whether self-supervised models trained on large speech databases can improve generalisation and deliver advance in spoof and deepfake detection performance.

9.1 Motivation

Given the fact that the training data used for ASVspoof challenges consists of spoofed utterances generated with a limited number of different attack algorithms (six in the case of the ASVspoof 2019 LA database), it may be difficult to improve generalisation without the use of more, external representative training data. On the other hand, the training of any supervised DNN front-end requires a large amount of bona fide and spoofed data, which is impractical to generate. This motivates the exploration of self-supervised pre-trained speech models as a CM front-end. Inspired by (i) SSL effectiveness in learning generalised feature representations for a variety of speech tasks [225–232], (ii) evidence that proper fine-tuning with modest amounts of labelled data can achieve state-of-the-art performance [233], (iii) encouraging results obtained from the use of self-supervised learning in anti-spoofing [234–236], and (iv) the appeal of one-class classification approaches [93, 237], we have explored the use of self-supervised learning to improve generalisation. Our hypothesis is that better representations trained on

diverse speech data at large-scale, even those learned for other tasks and initially using only bona fide data (hence one-class), may help to reduce over-fitting and improve reliability and domain robustness, particularly in the face of previously unseen spoofing attacks.

The key contributions of this work include:

- improved generalisation and domain robustness using a pre-trained, self-supervised speech model with fine-tuning;
- additional improvements using raw data augmentation showing complementary benefits to self-supervised learning;
- a new self-attention-based aggregation layer which brings complementary improvements.

There are several advantages of using a self-supervised DNN front-end for spoofing and deepfake detection tasks.

1. **Improved feature representation**: self-supervised front-ends can learn more robust and generalised representations of audio signals compared to manually processed hand-crafted features. This leads to better detection performance.
2. **Data efficiency:** self-supervised models can be trained on massive amounts of unlabelled data, which is available in abundance. This allows for an efficient training process, and can also lead to better generalisation to diverse spoofing attacks by fine-tuning using in-domain bona fide and spoof utterances.
3. **Reducing domain mismatch:** self-supervised models fine-tuned on in-domain data can be effective in addressing the domain-shift problem when in-domain data is limited.

9.2 Related work

Self-supervised learning (SSL) has recently gained growing attention in the speech research community. Many works in the literature have demonstrated that pre-trained SSL models can be adapted to multiple tasks using only a small amount of labelled data [193]. Recently, many self-supervised speech models have been proposed such as contrastive predictive coding (CPC) [229, 238], auto-regressive predictive coding [239], wav2vec [240], HuBERT [241, 242], wav2vec 2.0 [226, 243] and Wavlm [244]. These have shown promising results for various speech-related tasks, such as automatic speech recognition [226],

mispronunciation detection [245, 246], speaker recognition [247, 248] and emotion recognition [249]. Some studies have shown the benefits of using SSL for spoofing detection, e.g., Xie et al. [235] demonstrated the use of SSL with a Siamese network [250]. However, it is difficult to determine the specific benefits of SSL without comparative assessments using other representations or without exploring other methods such as data augmentation or domain mismatch.

Recent works have also investigated the use of self-supervised models such as wav2vec 2.0 [226] as a feature extractor (front-end) for spoofing detection. Martin-Donas et al. [251] used a SSL wav2vec 2.0 model as a feature extractor and proposed a method for the normalisation of encoded representations produced by the transformer layers to improve detection performance. Wang et al. [236] compared the use of different SSL front-ends and back-end architectures and demonstrated the importance of fine-tuning SSL models for spoofing detection. They showed a relative reduction in EER of 68% and 79% for the ASVspoof 2021 LA and DF databases respectively. However, they did not investigate the use of data augmentation to further improve domain robustness. Eom et al. [252] proposed a transfer learning approach based on wav2vec 2.0 with a variational information bottleneck (VIB) to extract more generalised information. Cai et al. [230] used an iterative self-supervised pre-training method to train a front-end model and showed that the learned features generalise better across different databases and also improve performance. In this work, we explore the use of the wav2vec 2.0 XLS-R model [253] as a CM front-end to learn more generalised feature representations. We further investigate the effectiveness of our proposed method using simple raw data augmentation technique called RawBoost described in Chapter 8, as well as a more sophisticated classifier, which brings complementary improvements.

9.3 Self-supervised front-end

In this section, we describe the replacement of the conventional sinc-layer front-end shown in Figure 9.1-(a) with the self-supervised wav2vec 2.0 front-end illustrated in Figure 9.1-(b). We describe both pre-training and fine-tuning processes to support downstream spoofing detection, both illustrated in Figure 9.2.

9.3.1 wav2vec 2.0 model

The pre-trained wav2vec 2.0 model is used to extract an output sequence of feature representations $o_{1:N}$ from the raw input waveform $x_{1:L}$, where L is the number of samples. As shown in Figure 9.2, the wav2vec 2.0 model consists of a convolutional neural network (CNN) and a transformer [153, 193] network. The former converts the input $x_{1:L}$ to a latent sequence $z_{1:N}$ whereas the latter transforms $z_{1:N}$ to an output sequence $o_{1:N}$. The ratio between L and N is dictated by the CNN stride

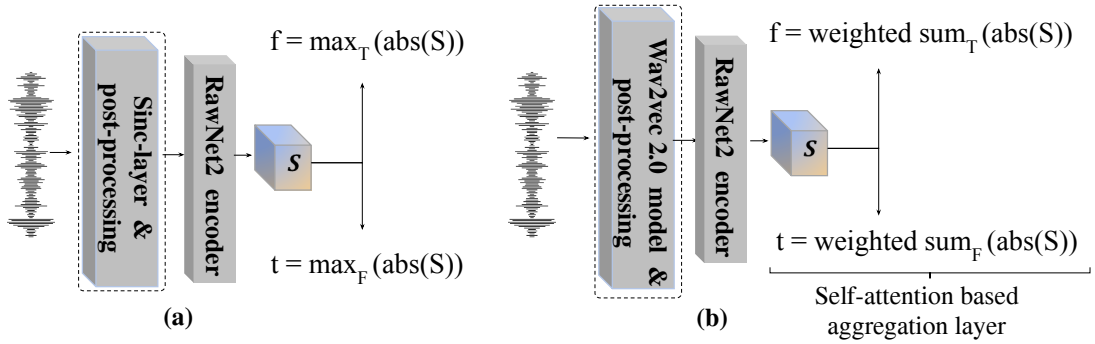


Figure 9.1: Front-end systems: (a) the baseline sinc-layer front-end; (b) the wav2vec 2.0 front-end.

of 20 ms (the default setting).

9.3.2 Pre-training

An illustration of the pre-training procedure following [226] is illustrated to the left in Figure 9.2. Latent representations $z_{1:N}$ are quantised to representations $q_{1:N}$. Some portion of the latent representation $z_{1:N}$ is then masked and fed to the transformer which builds new context representations $c_{1:N}$. A contrastive loss for each masked time step n is then computed to measure how well the target q_n can be identified from among a set of distractors (i.e., $q_{n'}$ sampled from the other masked time steps where $n' \neq n$) given the corresponding context vector c_n. All experiments were performed with the wav2vec 2.0 XLS-R (0.3M parameters) model [253]. We followed the example in the Fairseq project toolkit [254] to extract feature representations using the pre-trained self-supervised model wav2vec 2.0.[1]

9.3.3 Fine-tuning

Since pre-training is performed with only bona fide data (with no spoofed data), as per [236], fine-tuning using in-domain bona fide and spoofed training data is necessary in order to perform spoofing detection. Our hypothesis is that fine-tuning will prevent over-fitting and hence promote better generalisation to previously unseen attacks and different domains. For *all* experiments presented in this work, including those related to the ASVspoof 2021 LA dataset and the ASVspoof 2021 DF dataset, fine-tuning is performed using the ASVspoof 2019 LA training partition only. Whereas the 2021 LA data contains codec and transmission variation and

[1] https://github.com/pytorch/fairseq/tree/main/examples/wav2vec

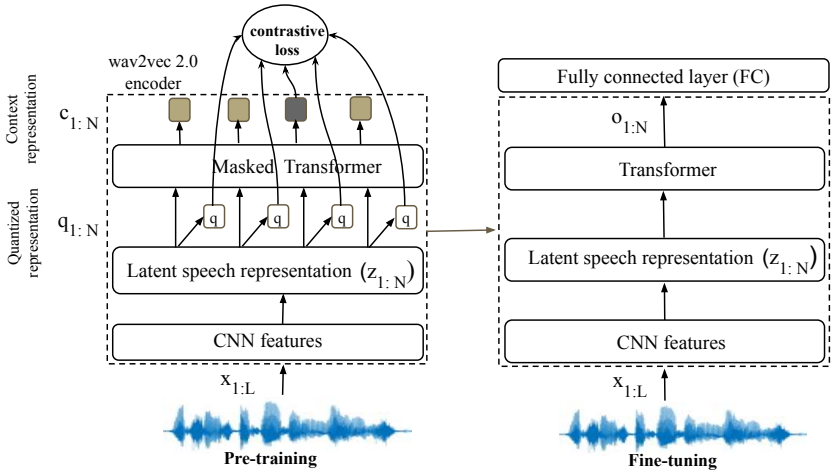


Figure 9.2: An overview of the pre-training and fine-tuning of the wav2vec 2.0 model.

the 2021 DF data contains compression variation, the 2019 LA data used for fine-tuning contains neither. During fine-tuning, the pre-trained wav2vec 2.0 XLS-R model is jointly optimised with the AASIST CM via back-propagation [156] using the ASVspoof 2019 LA training partition as illustrated in Figure 9.2. Fine-tuning is performed using a weighted cross-entropy objective function to minimise the training loss. In contrast to pre-training, input masking is not applied to latent representation $z_{1:N}$ during fine-tuning. Additionally, we add a fully connected layer on top of the wav2vec 2.0 transformer encoder output $o_{1:N}$ in order to reduce the output feature dimension (top-right of Figure 9.2).

9.4 Self-attention based aggregation layer

Attention-based pooling layers, such as self-attentive pooling (SAP) and attentive statistical pooling (ASP) [255] have been shown to be beneficial to the aggregation of frame-level features and the extraction of embeddings for speaker recognition and verification [141, 169, 256–258]. We have also found that the introduction of a 2-D self-attention based aggregation layer between the front-end and back-end helps to further improve spoofing detection performance. This new self-attentive aggregation layer is used to extract more attentive/relevant spectral and temporal features. It aggregates and assigns higher attention weights through weighted summation to the most discriminative temporal and spectral features. We generate 2-D attention maps (an attention weight matrix) using a 2-D convolutional (conv2d) layer with *one* kernel-size, rather than a conventional conv1d-based attention map applied to a single domain.

Weights are derived from the high-level feature representations S processed by a conv2d layer followed by an activation & batch normalisation layer, a 2-D convolutional layer, and a softmax layer to normalised the weights:

$$W = \textbf{Softmax}(\text{conv2d}(\text{BN}(\text{SeLU}(\text{conv2d}(\text{S}))))), \tag{9.1}$$

where conv2d($\cdot$) denotes the 2-D convolution operation with scaled exponential linear unit SeLU($\cdot$) as the activation function [152], and where BN is batch normalisation [175]. Temporal and spectral feature representations are then extracted using the self-attentive aggregation layer according to:

$$\text{t} = \sum_{F} \text{S} \odot W, \tag{9.2}$$

$$\text{f} = \sum_{T} \text{S} \odot W, \tag{9.3}$$

where $\odot$ denotes element-wise multiplication. $W \in \mathbb{R}^{F \times T}$ is the 2-D attention normalised learnable weight matrix used in the self-attentive aggregation layer to calculate the weighted *sum* of the feature representation S across time and frequency domains. A summary of the wav2vec 2.0 front-end and downstream AASIST model configurations is presented in Table 9.1.

9.5 Implementation details

Experiments were performed using the ASVspoof 2021 LA and DF evaluation databases described in Section 2.1. As per the ASVspoof 2021 Challenge rules [199, 221], we used the ASVspoof 2019 LA training data to fine-tune the spoofing CM and used development data for validation. We use AASIST model as a baseline system described in Chapter 7. Raw input waveforms are cropped or concatenated giving segments of approximately 4 seconds duration (64,600 samples). For all experiments, we used sinc-layer, fixed wav2vec 2.0 (without fine-tuning), and fine-tuned wav2vec 2.0 model as a front-ends. For SSL experiments, sinc-layer front-end illustrated in Figure 9.1-(a) is replaced with the wav2vec 2.0 front-end shown in Figure 9.1-(b). As described in Section 9.3, wav2vec 2.0 front-end output $o_{1:N}$ is fed to a RawNet2-based encoder which is used to learn higher-level feature representations S. AASIST baseline extracts temporal and spectral representations t and f from S using a max-pooling operation to construct input graph representation similar to described in Section 7.1. Whereas a self-attentive aggregation layer described in Section 9.4 was found to be effective for all front-ends. Temporal and spectral representations are then fed to the AASIST model to obtain a two-class prediction (bona fide and spoofed).

Table 9.1: The wav2vec 2.0 and AASIST model architecture. Dim. refer to (Channels, Frequency, Time). Batch normalisation (BN) and scaled exponential linear unit (SeLU), beneath the dotted line, are applied to RawNet2 encoder output.

	Layer	**Input**:64600 samples		**Output shape**
	Data-aug	RawBoost		(64600)
	SSL	wav2vec 2.0		(201,1024) (T,F)
	front-end	FC (fine-tuning)		(201,128)
		transpose		o=(128,201) (F,T)
	post-	add channel		(1,128,201)
	processing	Maxpool-2D(3)		(1,42,67)
		BN & SeLU		
RawNet2 encoder	Res-block	Conv-2D((2,3),1,32) BN & SeLU Conv-2D((2,3),1,32)	×2	(32,42,67)
	Res-block	Conv-2D((2,3),1,64) BN & SeLU Conv-2D((2,3),1,64)	×4	(64,42,67)
		BN & SeLU		
	Spectral-attention			Temporal-attention
	Self att. agg. layer ($\mathbf{f}$)=(64, 42)			Self att. agg. layer ($\mathbf{t}$)=(64, 67)
	$\mathcal{G}_{\mathrm{f}}$ = (64(d_{f}),21(N_{f}))			$\mathcal{G}_{\mathrm{t}}$ = (64(d_{t}),33(N_{t}))
	hetero. graph ($\mathcal{G}_{\mathrm{ft}}$)	HS-GAL		(64(d_{ft}),54(N_{ft}))
	HS-GAL→HS-GAL, stack node (32,26), (32,)			HS-GAL→HS-GAL, stack node (32,26),(32,)
	MGO ($\mathcal{G}_{FT}$)	element-wise max.		(32,26), (32,)
	readout	node-wise max. and avg., concat.		(160,)
	Output	FC(2)		2

The large wav2vec 2.0 (XLSR) model contains 24 transformer layers with model dimension 1024 and 16 attention heads. As illustrated to the right of Figure 9.2, a fully connected layer is applied to the output of the wav2vec 2.0 front-end to reduce the output dimensions from 1024 to 128. For the fixed SSL front-end (no fine-tuning), CM is trained in the same manner as the AASIST baseline. Graph pooling is applied in the AASIST model with an empirically chosen pooling ratio of $k = 0.5$ for spectral and temporal graphs. For model training, we used the standard Adam optimiser [155] with a fixed learning rate of 0.0001 with a batch size of 24 for experiments with baseline. Since SSL fine-tuning demands high GPU computation, experiments with wav2vec 2.0 front-end were performed with a smaller batch size of 14 and a lower learning rate of 10^{-6} to avoid model over-fitting. The proposed SSL-based CM is fine-tuned on ASVspoof 2019 LA training data to minimise a weighted cross entropy (WCE) loss function.

We applied RawBoost data augmentation (DA) in the same fashion and using the same configuration as described in Section 8.3 and with parameters reported in Table 8.2, using linear and non-linear convolutive noise and impulsive signal-dependent additive noise. These augmentation strategies suit the convolutive and device-related noise sources that characterise typically telephony applications. In contrast, for the DF database, DA works best using stationary signal-independent additive noise, which matches better the effects of audio compression [222] applied in generating the DF database. The best model was selected according to the minimum validation loss for the ASVspoof 2019 development data. All other hyperparameters are the same for both front-ends which are jointly optimised with the AASIST back-end classifier using back-propagation [156]. The performance of spoofing model can vary significantly with different random seeds as reported in [91]. In order to examine the variability in model performance, we performed each experiment with three runs using different random seeds to initialise the network weights (except for the pre-trained SSL front-end). All models were trained for 100 epochs. The model scripts and checkpoints are available in an open-source implementation.[2]

9.6 Results

In this section, we present six sets of experiments. The first is a comparison of each front-end in terms of performance for the ASVspoof 2021 LA database. The second and third assess the complementary benefits coming from the new self-attentive aggregation (SA) layer and RawBoost DA. The fourth is an assessment performed on the ASVspoof 2021 DF database. The fifth is a cross-databases evaluation

[2] https://github.com/TakHemlata/SSL_Anti-spoofing

Table 9.2: Pooled EER and pooled min t-DCF results for the ASVspoof 2021 LA database evaluation set, for the sinc-layer, wav2vec 2.0 fixed, and wav2vec 2.0 fine-tuned front-ends. Results are the best (average) obtained from three runs of each experiment with different random seeds. SA: Self-attentive aggregation layer; DA: Data augmentation.

front-end	SA	DA	Pooled EER	Pooled min t-DCF
sinc-layer	×	×	11.47 (11.95)	0.5081 (0.5139)
wav2vec 2.0 (fixed)	×	×	9.26 (12.46)	0.4097 (0.4293)
wav2vec 2.0 (fine-tuned)	×	×	6.15 (6.46)	0.3577 (0.3587)
sinc-layer	✓	×	8.73 (11.61)	0.4285 (0.5203)
wav2vec 2.0 (fixed)	✓	×	8.16 (10.24)	0.3897 (0.4058)
wav2vec 2.0 (fine-tuned)	✓	×	4.48 (6.15)	0.3094 (0.3482)
sinc-layer	×	✓	6.0 (6.18)	0.3532 (0.3583)
wav2vec 2.0 (fixed)	×	✓	7.32 (7.71)	0.3418 (0.3607)
wav2vec 2.0 (fine-tuned)	×	✓	1.19 (1.39)	0.2175 (0.2236)
sinc-layer	✓	✓	7.65 (7.87)	0.3894 (0.3960)
wav2vec 2.0 (fixed)	✓	✓	7.79 (9.05)	0.3407 (0.3608)
wav2vec 2.0 (fine-tuned)	✓	✓	**0.82 (1.00)**	**0.2066 (0.2120)**

assessment, whereas the last is a comparative assessment using a simplified CM solution.

9.6.1 Front-end comparison

Results for the AASIST baseline with the sinc-layer front-end (Figure 9.1-(a)) and the same system with the wav2vec 2.0 fixed and fine-tuned front-end (Figure 9.1-(b)) are presented in the first three rows of Table 9.2. These systems use neither SA layer nor DA. The baseline EER of 11.47% is high and shows that the system is not robust to the codec and transmission variability which characterises the ASVspoof 2021 LA dataset. The same system using the wav2vec 2.0 fixed and fine-tuned front-end delivers an EER of 9.26% and 6.15% respectively. Figure 9.3 depicts an attack and codec condition analysis for the LA database, where Figure 9.3-(a) shows a breakdown in the EER decomposed over spoofing attacks and Figure 9.3-(b) shows a breakdown EER decomposed over codecs conditions. Experimental results demonstrate that our wav2vec 2.0 front-end with fine-tuning consistently outperforms the baseline across different spoofing attacks and codec conditions.

9.6.2 Self-attentive aggregation layer

Results for the same two front-end variants but using the SA layer introduced in Section 9.4 are presented in rows 4–6 of Table 9.2. In all cases, the EER drops

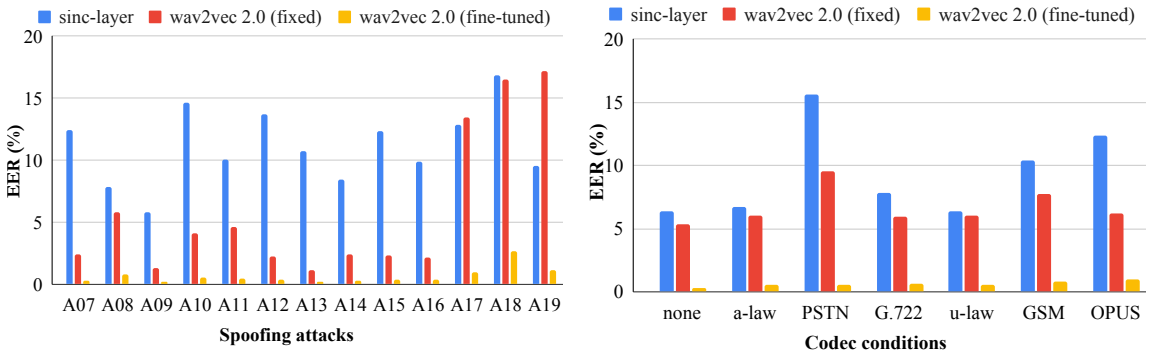


Figure 9.3: Decomposed EERs (a) across spoofing attacks, and (b) across codec conditions for ASVspoof 2021 LA dataset.

substantially, to 8.73% for the sinc-layer front-end, to 8.16% for the wav2vec 2.0 fixed front-end, and to 4.48% for the wav2vec 2.0 fine-tuned front-end. In this case, the wav2vec 2.0 fine-tuned front-end is responsible for a relative improvement of almost 50%.

9.6.3 Data augmentation

Results for the same systems, but using the RawBoost DA introduced in Chapter 8, are presented in rows 7–9 of Table 9.2. In all three cases using DA the EER drops substantially, to 6% for the sinc-layer front-end, to 7.32% for the wav2vec 2.0 fixed front-end, and to 1.19% for the wav2vec 2.0 fine-tuned front-end. The wav2vec 2.0 fine-tuned front-end with DA showed a substantial improvement in EER of 81%. Figure 9.4 shows codec performance with and without RawBoost DA using the wav2vec 2.0 fine-tuned front-end. The use of RawBoost DA further enhances the detection performance, especially for the most difficult (PSTN) telephony condition. These results demonstrate the effectiveness of RawBoost DA in more realistic and challenging telephony scenarios.

We also performed experiments to leverage the benefits of SA and DA together. Results for the same systems, both with the SA layer, and now also with DA, are shown in rows 10–12 of Table 9.2. DA reduces the EER only marginally from 8.73% to 7.65% in case of the sinc-layer front-end. Its effect is more pronounced when using the fine-tuned wav2vec 2.0 front-end for which the EER decreases from 4.48% to 0.82%. This result corresponds to a relative improvement of almost 90% when compared to the baseline EER of 7.65%. To the best of our knowledge, this was the lowest EER reported for the ASVspoof 2021 LA database at the time of writing.

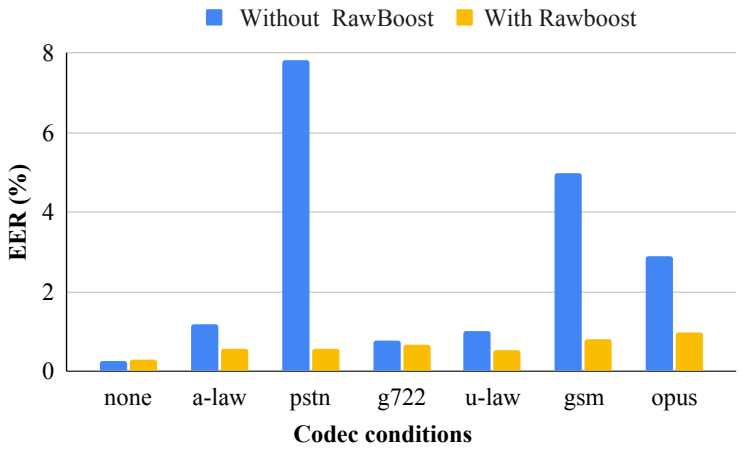


Figure 9.4: Performance comparisons of fine-tuned wav2vec 2.0 front-end with and without RawBoost data augmentation.

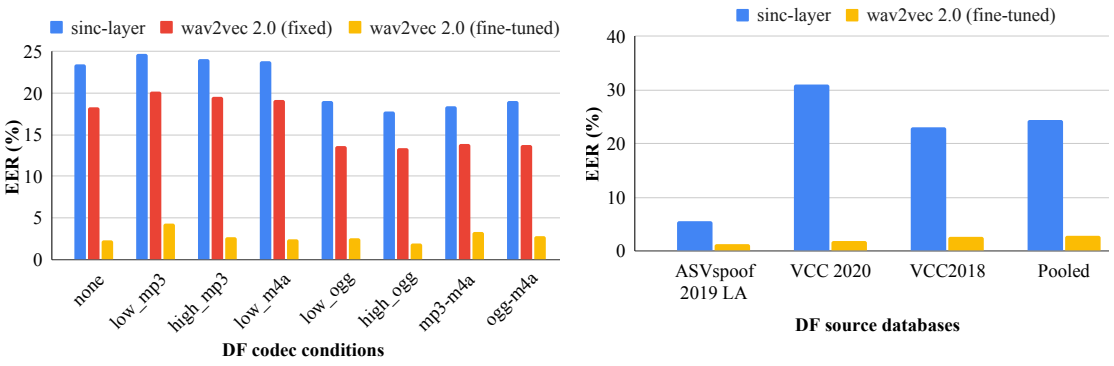
9.6.4 Deepfake results

Results for the same experiments, but for the ASVspoof 2021 DF database, are shown in Table 9.3. While neither SA, nor DA improve upon the baseline EER of 21.06%, consistency improvements are obtained for the wav2vec 2.0 fine-tuned front-end for which the EER drops from 7.69% to 2.85% using both SA and DA. To the best of our knowledge, this was the lowest EER reported for the ASVspoof 2021 DF database at the time of writing. These results, while determined with the same wav2vec 2.0 front-end used for LA experiments, relate to a DA strategy optimised for the DF database (stationary signal-independent additive noise – see Section 8.3). A component of the DF database originates from multiple data resources [199] including spoofed utterances generated with more than 100 diverse attack algorithms.

Figure 9.5-(a) depicts a breakdown of EERs (%) across evaluation codec conditions. Results show that with proper fine-tuning, the SSL front-end brings substantial improvements in detection performance over the baseline system. With the ASVspoof 2019 LA training data containing neither codec nor different compression effects, results show that the use of fine-tuned SSL models leads to consistent improvements in generalisation, here being previously unseen spoofing attacks. Results for the DF database show that the benefit extends also to the case of domain mismatch. As shown in Figure 9.5-(b), the SSL front-end also outperforms the baseline on in-domain ASVspoof 2019 LA evaluation data and out-of-domain VCC 2018 and 2020 source data as well. We further checked the impact of different vocoders used in VCC 2018 [45] and 2020 [46] subsets on DF performance.

Table 9.3: As for Table 9.2 except for the ASVspoof DF database, evaluation set. Since there is no ASV in the DF scenario, there are no min t-DCF results.

Front-end	SA	DA	Pooled EER
sinc-layer	×	×	21.06 (22.11)
wav2vec 2.0 (fixed)	×	×	14.22 (16.46)
wav2vec 2.0 (fine-tuned)	×	×	7.69 (9.48)
sinc-layer	✓	×	23.22 (25.08)
wav2vec 2.0 (fixed)	✓	×	19.98 (21.56)
wav2vec 2.0 (fine-tuned)	✓	×	4.57 (7.70)
sinc-layer	×	✓	16.62 (18.64)
wav2vec 2.0 (fixed)	×	✓	16.05 (17.01)
wav2vec 2.0 (fine-tuned)	×	✓	3.64 (3.98)
sinc-layer	✓	✓	24.42 (25.38)
wav2vec 2.0 (fixed)	✓	✓	19.98 (20.35)
wav2vec 2.0 (fine-tuned)	✓	✓	**2.85 (3.69)**



(a) (b)

Figure 9.5: DF decomposed EER (a) across different codecs ; (b) across different ASVspoof 2019 LA, VCC 2020 and VCC 2018 source databases.

Table 9.4 shows a breakdown in the EER for VCC 2018 and 2020 data according to the type of codec and vocoder (breakdown results correspond to the last row in Table 9.3). Results indicate that, for any given codec, neural AR vocoders yield worse EERs than waveform concatenation and traditional vocoders.

Table 9.4: Decomposed EERs using wav2vec 2.0 front-end for VCC 2018/2020 subset of the DF database, according to the codecs and the vocoder type used in voice conversion.

Vocoder type	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**pooled**
Unknown	1.99	4.30	2.65	2.10	2.23	1.27	2.66	2.14	2.45
Wav. Concat	2.28	5.84	3.35	2.09	2.23	1.50	2.96	2.52	2.85
Neural non-AR	1.56	3.33	2.02	1.65	1.62	1.00	2.05	1.57	1.84
Neural AR	3.45	5.96	3.79	3.75	3.67	2.92	4.49	3.79	4.05
Traditional	1.22	2.72	1.83	1.57	2.35	1.57	3.01	2.28	2.15

9.6.5 Cross-database evaluation

To evaluate the generalisation capability of our proposed method, we tested it on multiple test sets. In cross-database evaluation settings, the model was trained on the ASVspoof 2019 LA training set and evaluated on multiple test sets, including the ASVspoof 2015 [15], 2019 LA [53], and 2021 LA and DF test sets [199]. Our results, shown in Table 9.5, indicate that the SSL front-end features have excellent transferability in cross-database settings. This suggests that the CM model is able to generalise well to different test sets and domains as well. In particular, the model performed well on the more challenging ASVspoof 2021 LA and DF tasks, which measure the model's generalisability to unknown attacks and mismatched domains. In conclusion, using a well-trained, fine-tuned SSL front-end can significantly improve the generalisation of a model and help it to perform better on a wide range of attacks.

Table 9.5: Performance comparisons of single CM systems in terms of pooled EER (%) across cross-evaluation databases, namely, ASVspoof 2015, 2019 and 2021 LA and DF databases.

Systems	**front-end**	**2015 LA**	**2019 LA**	**2021 LA**	**2021 DF**
Wang et al. [236]	wav2vec 2.0	**0.24**	2.31	7.18	6.18
Eom et al. [252]	wav2vec 2.0	1.52	0.40	4.92	-
Wang et al. [259]	wav2vec 2.0	-	**0.20**	3.73	3.28
Martin et al. [251]	wav2vec 2.0	-	-	3.54	4.98
Wang et al. [260]	wav2vec 2.0	0.59	0.21	3.30	4.12
Ours: Proposed	wav2vec 2.0	**0.24**	0.21	**0.82**	**2.85**

Table 9.6: Pooled EER and pooled min t-DCF (LA only) results for the ASVspoof 2021 LA and DF databases, evaluation sets, using their respective optimised RawBoost data augmentation (DA) strategies and the simplified back-end.

Front-end	DA	Database	Pooled EER	Pooled min t-DCF
wav2vec 2.0	✓	LA	1.19	0.2175
wav2vec 2.0	✓	DF	4.38	-
wav2vec 2.0	×	LA	7.15	0.3830
wav2vec 2.0	×	DF	9.55	-

9.6.6 Simplified countermeasure solution

The last set of experiments were performed in order to investigate the relative importance of the more sophisticated AASIST back-end and to determine whether the improvements in generalisation are obtained also for a simpler CM solution. We removed the RawNet2-based encoder and replaced AASIST back-end with a simple back-end comprising a max-pooling layer, a single graph module layer and a linear layer. Results for both ASVspoof 2021 LA and DF databases using optimised DA strategies for each are shown in Table 9.6. Results are not as good as for the more sophisticated AASIST back-end. However, LA and DF results of 1.19% and 4.38% for the simple CM show that competitive EERs can nonetheless be obtained using the fine-tuned wav2vec 2.0 front-end even with relatively less complex networks and that the benefits to generalisation are still complementary to those of DA. However, when the SSL front-end was fine-tuned, the choice of the back-end has less impact on performance. Even the use of a simple graph layer as a back-end achieves competitive performance for the ASVspoof 2021 LA and DF databases.

9.7 Summary

The work presented in this chapter demonstrates that a well-trained and fine-tuned front-end can substantially improve generalisation, even when learned initially using massive quantities of only *bona fide* utterances. When combined with a new SA layer and RawBoost DA technique, the SSL front-end outperforms a conventional sinc-layer-based front-end by delivering up to a 90% relative reduction in the EER for the LA task and up to an 88% relative reduction for a domain mis-matched DF task. Extensive experiments show that SSL-based front-end fine-tuned on ASVspoof 2019 training data performs well on 2015, 2021 LA and 2021 DF test sets and also converges faster than conventional front-ends. These results demonstrate that the use of larger and more diverse training data helps to improve the generalisability of CMs in-the-wild scenarios.

Chapter 10

Conclusions and Future Directions

In this chapter, we present a summary of the contributions and findings of the work presented in this thesis. This material is reported in Section 10.1. Some potential directions for future research are discussed in Section 10.2.

10.1 Summary

As described in Chapter 1, this thesis aimed to develop robust countermeasure (CM) systems to improve detection performance for a wide range of spoofing attacks and further reduce the performance gap between the best ensemble and best single systems. We developed a set of spoofing CMs which brought advances in the state-of-the-art as judged from experiments performed using the ASVspoof 2019 and ASVspoof 2021 LA and DF databases. Figure 10.1 depicts detection error trade-off (DET) profiles for single system CMs proposed in this thesis, in addition to the set of ASVspoof 2019 LA challenge submissions (gray profiles). Highlighted profiles in Figure 10.1 correspond to the single systems proposed in this thesis, the best ensemble (T05) and best single (T45) systems from among the ASVspoof 2019 LA challenge entries, and the best B02 challenge baseline (LFCC-GMM). Figure 10.1 shows that the proposed graph neural network (GNN) based solutions, RawGAT-ST and AASIST, reduce the performance gap to the best ensemble solution (T05) by a substantial margin. The self-supervised learning (SSL) based CM system significantly outperforms all other systems. The SSL-based CM is a new, fully-reproducible, efficient, state-of-the-art solution for spoofing and deepfake detection. We note that the best ensemble T05 system remains unreproducible to date. We also acknowledge that this comparison is between evaluation and post-evaluation performance.

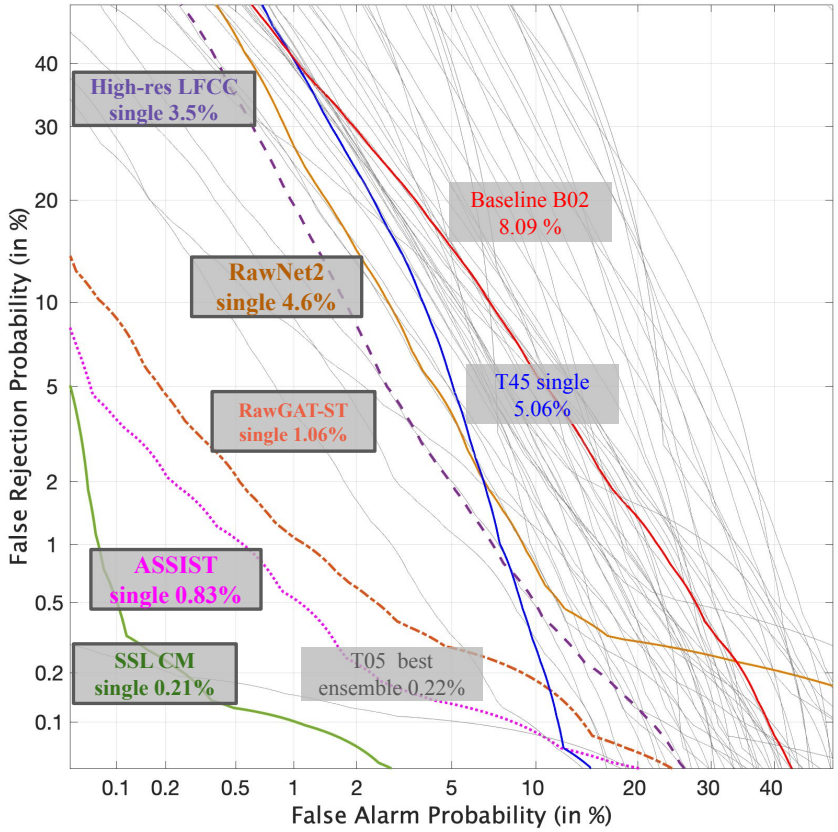


Figure 10.1: DET profiles for proposed single systems along with ASVspoof 2019 LA challenge submissions results. Proposed single CM systems (solid box). All grey profiles from the ASVspoof 2019 LA challenge entries (ensemble systems).

Thesis contributions are summarised in the following. Chapters 4 to 7 focus on the development of novel spoofing CMs. Chapters 8 and 9 focus on generalisation techniques to improve CM performance in more realistic scenarios.

Chapter 3 presents an explainability study of constant Q cepstral coefficients (CQCCs), one of the most popular spoofing CM front-ends when this work began. Reported, is an investigation of why the CQCC front-end works so reliably in detecting some spoofing attack, but why it fails to generalise to others. Through a sub-band analysis of the CQCC-GMM CM system, we found that CQCCs without resampling perform well in detecting spoofing artefacts when they are located at low frequencies. Linear sampling (resampling) shifts the emphasis to higher frequencies so that spoofing artefacts at similarly high frequencies are emphasised and captured reliably. This work showed that no single silver bullet works well for a diverse range of spoofing attacks; different spoofing attacks produce artefacts at

different parts of the spectrum and these can only be detected reliably when the front-end emphasises information in the same frequency bands. These findings may explain why classifier fusion has proven to be so important to generalisation.

In **Chapter 4**, we introduced a non-linear ensemble approach comprising a set of sub-band CMs, each tuned to detect artefacts in different sub-bands. The proposed high spectral resolution front-end outperforms the baseline by a large margin. Furthermore, the non-linear fusion of sub-band CMs significantly improves detection performance.

In **Chapter 5**, we introduced an end-to-end RawNet2 CM system which learns representative features automatically. The end-to-end system operates directly on raw waveform inputs without any pre-processing transformation, streamlining training and evaluation. Results confirm that the end-to-end classifier is capable of learning spoofing cues that are different to those captured with traditional front-end features. Traditional hand-crafted front-end features and the features learned directly from raw waveforms are hence complementary, and further improvements can be achieved by combining these different feature representations. The RawNet2 model was adopted as a baseline for the most recent ASVspoof 2021 challenge and outperformed other systems that use traditional hand-crafted features.

Inspired by the effectiveness of the end-to-end RawNet2 model, in **Chapter 6** we introduced an end-to-end spectro-temporal graph attention network, called RawGAT-ST. It also operates directly upon raw waveform inputs and concurrently learns the relationship between discriminative cues in both spectral and temporal domains. Results show that the RawGAT-ST CM generalises well to unseen attacks and achieved the lowest reported EER for the ASVspoof 2019 LA database at the time of publication. **Chapter 7** presents an extension to the RawGAT-ST model, which introduces a heterogeneous graph attention layer leading to an integrated spectro-temporal graph attention network, named AASIST. AASIST incorporates a new heterogeneity-aware attention mechanism, a max-graph operation, and an additional stack node, which facilitates the concurrent modeling of heterogeneous temporal and spectral graph representations. The work also analyses the impact of joint spectro-temporal attention upon the most difficult-to-detect spoofing attacks. We demonstrated the benefit of self-attention in learning the relationship between spectral and temporal cues for spoofed speech detection.

Further contributions to improve CM generalisation and domain robustness

in real-world scenarios, are reported in Chapters 8 and 9. In **Chapter 8**, we propose a novel data augmentation technique, *RawBoost*, which can be used to introduce nuisance variation stemming from unknown encoding, transmission and compression to training data thereby reducing domain mismatch between training and testing data. RawBoost is data, application and model agnostic. The RawBoost algorithms introduce non-linearity into the input utterances, which effectively aligns with the neural networks and enables it to learn more complex patterns and information from the training data. While this thesis demonstrates its application to improve spoofing and deepfake detection performance, it might have application to other related classification tasks, e.g. automatic speaker verification or automatic speech recognition.

Chapter 9 explores a self-supervised based front-end which is trained on a large quantity of diverse speech data. Experiments with the most challenging ASVspoof 2021 LA and DF databases show that the use of RawBoost data augmentation with a SSL wav2vec 2.0 front-end brings substantial improvements in performance showing that a well-trained, fine-tuned front-end, even when trained initially using massive quantities of only *bona fide* utterances, can improve generalisation. Our new CM solution for logical access and deepfake detection improved further upon the state-of-the-art. Its performance remains to be bettered. Data augmentation not only helps prevent over-fitting, but also improves model robustness to different kinds of compression and channel variation. Our system shows the best performance for an unknown telephony condition (PSTN+VoIP) and reliable performance for different compression (ogg) for the DF database. Results indicate the potential gain in performance which can be obtained with the use of additional, diverse external training data. This might suggest that the relaxed training policy which allows for the use of larger, more complex CM models trained using external data might be worth adopting for future ASVspoof evaluations.

10.2 Future directions

As research efforts in spoofing and deepfake detection continue to grow, several promising future research directions have emerged. In the following, we outline four major directions that we believe hold great potential for advancing CM effectiveness in real-world scenarios:

1. **Background and real-world noises:** Past and current spoofing detection studies have relied on ASVspoof databases which contain clean data without background noise such as babble noise, volvo noise and cafe noise. It is important to determine whether graph neural networks and self-supervised learning-based CM systems are effective in such more realistic conditions. As

the literature [219, 220] shows, CM performance degrades significantly in the presence of background noise. However, attackers can easily deceive system by introducing real-world noises, hence there is a need for further research to develop noise-robust CMs. Future challenges should take the additive background noise into account. This work might also help to address the machine learning shortcut issue connected to spoofing detection with the use of information in non-speech intervals, as reported in [112, 261, 262].

2. **Diversity in database collection:** To better understand the challenges posed by new and unpredictable attacks in real-world settings, more comprehensive, larger databases are required. Some database initiatives in this direction already exist, for e.q. Fake or Real (FoR) [263] and SYN-SPEECHDDB [264] databases for synthetic speech detection. Countermeasures tested on the recent ASVspoof 2021 DF database show a performance gap between progress and evaluation subsets, indicating model over-fitting to known attacks in the training data. To mitigate this issue, training with large and diverse data is essential, as discussed in Chapter 9, and as shown from the exploration of a self-supervised front-end trained on extensive data. Increasing the diversity of bona fide and spoofed utterances by including a greater number of different spoofing algorithms, diverse languages, different accents and data collected from more speakers might help to improve generalisation in real-world scenarios.

3. **Adversarial attacks:** Investigating the impact of adversarial attacks on ASV and CM model decisions is a interesting research topic within the broader speech community [265, 266]. Some studies show the threat posed by adversarial attacks to reliable ASV [267, 268]. Adversarial examples are meticulously crafted speech samples that are difficult to distinguish from the original input samples, and often imperceptible to humans. Ensuring the robustness of ASV and CM models against such malicious attacks is crucial for the development of secure voice biometrics systems. Although some recent studies [267, 268] have focused on adversarial attacks primarily targeting the ASV systems, attacks on the CM are less common. However, since the reliability of security estimates is only as good as, the strength of the adversary model, the investigation of adversarial attacks that target both ASV and CM systems will be extremely important in the future.

4. **Joint audio-visual deepfake detection:** In recent years, researchers have proposed several deepfake detection algorithms to determine whether an audio frame or a visual frame is manipulated. Most of the research work primarily focuses on single-modality deepfake detection [88, 269–271]. Multi-modal deepfake detection in real-world scenarios remains a challenging task. Also,

it is very common in current deepfake detection tasks, that detectors can not obtain the number of modalities and forgery methods of deepfakes in advance. Therefore, in this case, it is interesting to explore graph neural network-based approaches to learn the correlations or relationships between audio-visual cues to improve the performance of deepfake detection in the wild.

Bibliography

[1] J. H. 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.

[2] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[3] 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.

[4] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–5.

[5] "ISO/IEC 30107. Information Technology – Biometric presentation attack detection," International Organization for Standardization, Geneva, Switzerland, Standard, 2016.

[6] "ISO/IEC FDIS 19795-1:2021. Information Technology – Biometric performance testing and reporting – Part 1: Principles and framework," International Organization for Standardization, Geneva, Switzerland, Standard, 2021.

[7] M. Faundez-Zanuy, M. Hagmüller, and G. Kubin, "Speaker verification security improvement by means of speech watermarking," *Speech communication*, vol. 48, no. 12, pp. 1608–1619, 2006.

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

[9] B. L. Pellom and J. H. Hansen, "Voice forgery using alisp," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, pp. 837–840.

[10] P. Patrick, G. Aversano, R. Blouet, M. Charbit, and G. Chollet, "Voice forgery using ALISP: indexation in a client memory," in *Proc. ICASSP*, 2005, pp. 17–20.

[11] T. Masuko, T. Hitotsumatsu, K. Tokuda, and T. Kobayashi, "On the security of HMM-based speaker verification systems against imposture using synthetic speech," in *Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999)*, 1999, pp. 1223–1226.

[12] J. Lindberg and M. Blomberg, "Vulnerability in speaker verification-a study of technical impostor techniques," in *Sixth European conference on speech communication and technology*, 1999.

[13] J. Villalba and E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems," in *European Workshop on Biometrics and Identity Management*. Springer, 2011, pp. 274–285.

[14] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *Proc. of International Speech Communication Association (INTERSPEECH)* , 2013, pp. 925–929.

[15] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilci, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 2037–2041.

[16] M. Sahidullah, T. Kinnunen, and C. Hanilci, "A comparison of features for synthetic speech detection," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 2087–2091.

[17] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[18] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[19] 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," in *Proc. Speech Synthesis Workshop (SSW)*, 2016, p. 125.

[20] M. Schröder, M. Charfuelan, S. Pammi, and I. Steiner, "Open Source Voice Creation Toolkit for the MARY TTS Platform," in *Proc. INTERSPEECH*, 2011, pp. 3253–3256.

[21] M. Pal, D. Paul, and G. Saha, "Synthetic speech detection using fundamental frequency variation and spectral features," *Computer Speech & Language*, vol. 48, pp. 31–50, 2018.

[22] I. Saratxaga, J. Sanchez, Z. Wu, I. Hernaez, and E. Navas, "Synthetic speech detection using phase information," *Speech Communication*, vol. 81, pp. 30–41, 2016.

[23] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[24] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[25] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "Wavenet vocoder with limited training data for voice conversion." in *Interspeech*, 2018, pp. 1983–1987.

[26] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *Proc. ICASSP*, 2009, pp. 3893–3896.

[27] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," in *2018 IEEE Proc. ICASSP*. IEEE, 2018, pp. 5279–5283.

[28] M. Sahidullah, H. Delgado, M. Todisco, T. Kinnunen, N. Evans, J. Yamagishi, and K.-A. Lee, "Introduction to voice presentation attack detection and recent advances," in *Handbook of biometric anti-spoofing*. Springer, 2019, pp. 321–361.

[29] C. B. Tan, M. H. A. Hijazi, N. Khamis, Z. Zainol, F. Coenen, A. Gani *et al.*, "A survey on presentation attack detection for automatic speaker verification systems: State-of-the-art, taxonomy, issues and future direction," *Multimedia Tools and Applications*, vol. 80, no. 21, pp. 32 725–32 762, 2021.

[30] "What is a deepfake? everything you need to know about the aipowered fake media." [Online]. Available: https://www.businessinsider.com/guides/tech/what-is-deepfake?IR=T

[31] B. Thormundsson, "Potential ai-enabled cyberattacks on companies worldwide 2021." [Online]. Available: https://www.statista.com/statistics/1235395/worldwide-ai-enabled-cyberattacks-companies/

[32] J. Damiani, "A voice deepfake was used to scam a CEO out of 243,000," *Forbes Magazine*, 2019. [Online]. Available: https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/

[33] T. Brewster, "Fraudsters cloned company director's voice in 35 million bank heist, police find," *Forbes, Editor's Pick*, vol. 14, 2021, https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=7dfbccf67559.

[34] Z. Almutairi and H. Elgibreen, "A review of modern audio deepfake detection methods: Challenges and future directions," *Algorithms*, vol. 15, no. 5, p. 155, 2022.

[35] Y. Rodríguez-Ortega, D. M. Ballesteros, and D. Renza, "A machine learning model to detect fake voice," in *Applied Informatics: Third International Conference, ICAI 2020, Ota, Nigeria, October 29–31, 2020, Proceedings 3*. Springer, 2020, pp. 3–13.

[36] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen, and T. Matsui, "Voice liveness detection for speaker verification based on a tandem single/double-channel pop noise detector." in *Odyssey*, vol. 2016, 2016, pp. 259–263.

[37] M. Sahidullah, D. A. L. Thomsen, R. G. Hautamäki, T. Kinnunen, Z.-H. Tan, R. Parts, and M. Pitkänen, "Robust voice liveness detection and speaker verification using throat microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 44–56, 2017.

[38] M. R. Kamble, H. B. Sailor, H. A. Patil, and H. Li, "Advances in anti-spoofing: from the perspective of ASVspoof challenges," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.

[39] A. Khan, K. M. Malik, J. Ryan, and M. Saravanan, "Voice spoofing countermeasures: Taxonomy, state-of-the-art, experimental analysis of generalizability, open challenges, and the way forward," *arXiv preprint arXiv:2210.00417*, 2022.

[40] 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 *Proc. INTERSPEECH*, 2017.

[41] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof2021: accelerating progress in spoofed and deep fake speech detection," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[42] H. Delgado, M. Todisco, N. Evans, M. Sahidullah, W. M. Liu, F. Alegre, T. Kinnunen, and B. Fauve, "Impact of bandwidth and channel variation on presentation attack detection for speaker verification," *International Conference of the Biometrics Special Interest Group (BIOSIG)*.

[43] G. Lavrentyeva, S. Novoselov, M. Volkova, Y. Matveev, and M. De Marsico, "Phonespoof: A new dataset for spoofing attack detection in telephone channel," in *Proc. ICASSP*, 2019, pp. 2572–2576.

[44] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2017, http://dx.doi.org/10.7488/ds/1994.

[45] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.

[46] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," *arXiv preprint arXiv:2008.12527*, 2020.

[47] A. Nautsch, X. Wang *et al.*, "ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, 2021.

[48] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019.

[49] M. Todisco, X. Wang *et al.*, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. INTERSPEECH*, 2019, pp. 1008–1012.

[50] ASVspoof 2019: the automatic speaker verification spoofing and countermeasures challenge evaluation plan. [Online]. Available: http://www.asvspoof.org/asvspoof2019/asvspoof2019_evaluation_plan.pdf

[51] H. Delgado, N. Evans, T. Kinnunen, K. A. Lee, X. Liu, A. Nautsch, J. Patino, M. Sahidullah, M. Todisco, X. Wang *et al.*, "ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," *arXiv preprint arXiv:2109.00535*, 2021.

[52] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "ASVspoof 2015: the first automatic verification spoofing and countermeasures challenge evaluation plan," in *Proc. IEEE Signal Process. Soc. Speech Lang. Tech. Committee Newslett.*, 2014.

[53] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, 2020.

[54] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch *et al.*, "ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild," *arXiv preprint arXiv:2210.02437*, 2022.

[55] N. Brümmer and E. de Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing," 2011.

[56] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Odyssey*, 2018.

[57] T. Kinnunen, H. Delgado *et al.*, "Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals," *IEEE/ACM Transactions on Audio Speech and Language Processing (TASLP)*, vol. 28, 2020.

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

[59] X. Wang and J. Yamagishi, "A practical guide to logical access voice presentation attack detection," in *Frontiers in Fake Media Generation and Detection*. Springer, 2022, pp. 169–214.

[60] A. Mittal and M. Dua, "Automatic speaker verification systems and spoof detection techniques: review and analysis," *International Journal of Speech Technology*, vol. 25, no. 1, pp. 105–134, 2022.

[61] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

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

[63] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *Proc. INTERSPEECH*, 2015, p. 2062–2066.

[64] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin, "STC anti-spoofing systems for the ASVspoof 2015 challenge," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5475–5479.

[65] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu, "Robust deep feature for spoofing detection—The SJTU system for ASVspoof 2015 challenge," in *Proc. INTERSPEECH*, 2016.

[66] X. Xiao, X. Tian, S. Du, H. Xu, E. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: the NTU approach for ASVspoof 2015 challenge." in *INTERSPEECH*, 2015, pp. 2052–2056.

[67] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis, "Development of CRIM system for the automatic speaker verification spoofing and countermeasures challenge 2015," in *Proc. INTERSPEECH*, 2016.

[68] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-end convolutional neural network-based voice presentation attack detection," in *2017 IEEE international joint conference on biometrics (IJCB)*, 2017, pp. 335–341.

[69] C. Zhang, C. Yu, and J. H. Hansen, "An investigation of deep-learning frameworks for speaker verification antispoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 684–694, 2017.

[70] X. Tian, X. Xiao, E. S. Chng, and H. Li, "Spoofing speech detection using temporal convolutional neural network," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[71] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc. Speaker Odyssey Workshop*, Bilbao, Spain, 6 2016.

[72] 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*, vol. 28, no. 4, pp. 357–366, 1980.

[73] L. D. Alsteris and K. K. Paliwal, "Short-time phase spectrum in speech processing: A review and some experimental results," *Digital signal processing*, vol. 17, no. 3, pp. 578–616, 2007.

[74] 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.

[75] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *Proc. ICASSP*, 2018.

[76] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[77] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[78] S. Cui, B. Huang, J. Huang, and X. Kang, "Synthetic speech detection based on local autoregression and variance statistics," *IEEE Signal Processing Letters*, vol. 29, pp. 1462–1466, 2022.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[80] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, 2018.

[81] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. International conference on machine learning (ICML)*, 2017, pp. 933–941.

[82] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[83] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 1033–1037.

[84] Q. Fu, Z. Teng, J. White, M. E. Powell, and D. C. Schmidt, "Fastaudio: A learnable audio front-end for spoof speech detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3693–3697.

[85] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, 2019.

[86] R. K. Das, J. Yang, and H. Li, "Long range acoustic features for spoofed speech detection." in *INTERSPEECH*, 2019, pp. 1058–1062.

[87] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Proc. Odyssey*, 2020.

[88] A. Luo, E. Li, Y. Liu, X. Kang, and Z. J. Wang, "A capsule network based approach for detection of audio spoofing attacks," in *Proc. ICASSP*, 2021, pp. 6359–6363.

[89] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," in *Proc. ICASSP*, 2021, pp. 6354–6358.

[90] J. Yang, H. Wang, R. K. Das, and Y. Qian, "Modified magnitude-phase spectrum information for spoofing detection," *IEEE/ACM TASLP*, vol. 29, pp. 1065–1078, 2021.

[91] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," in *Proc. INTERSPEECH*, 2021.

[92] F. Alegre, A. Amehraye, and N. Evans, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," in *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013, pp. 1–8.

[93] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[94] A. Pianese, D. Cozzolino, G. Poggi, and L. Verdoliva, "Deepfake audio detection by speaker verification," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2022, pp. 1–6.

[95] Y. Zhang, G. Zhu *et al.*, "An Empirical Study on Channel Effects for Synthetic Voice Spoofing Countermeasure Systems," in *Proc. INTERSPEECH*, 2021.

[96] A. Tomilov, A. Svishchev *et al.*, "STC Antispoofing Systems for the ASVspoof 2021 Challenge," in *Proc. ASVspoof 2021 Workshop*, 2021.

[97] T. Chen, E. Khoury *et al.*, "Pindrop labs' submission to the ASVspoof 2021 challenge," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[98] R. K. Das, "Known-unknown data augmentation strategies for detection of logical access, physical access and speech deepfake attacks: ASVspoof 2021," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[99] X. Chen, Y. Zhang *et al.*, "Ur channel-robust synthetic speech detection system for ASVspoof 2021," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[100] W. H. Kang, J. Alam *et al.*, "CRIM's system description for the ASVspoof 2021 Challenge," in *Proc. ASVspoof 2021 Workshop*, 2021.

[101] J. Cáceres, R. Font *et al.*, "The Biometric Vox System for the ASVspoof 2021 Challenge," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[102] X. Wang, X. Qin, T. Zhu, C. Wang, S. Zhang, and M. Li, "The dku-cmri system for the ASVspoof 2021 challenge: Vocoder based replay channel response estimation," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite) (INTERSPEECH satellite)*, 2021.

[103] Y. Zhao, R. Togneri *et al.*, "Replay anti-spoofing countermeasure based on data augmentation with post selection," *Computer Speech & Language*, vol. 64, 2020.

[104] R. K. Das, J. Yang, and H. Li, "Data augmentation with signal companding for detection of logical access attacks," in *Proc. ICASSP*, 2021, pp. 6349–6353.

[105] X. Chen, Y. Zhang *et al.*, "UR Channel-Robust Synthetic Speech Detection System for ASVspoof 2021," in *Proc. ASVspoof 2021 Workshop*, 2021.

[106] A. Fathan, J. Alam, and W. H. Kang, "Mel-spectrogram image-based end-to-end audio deepfake detection under channel-mismatched conditions," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[107] D. S. Park, W. Chan *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. INTERSPEECH*, 2019.

[108] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 10 2015. [Online]. Available: http://arxiv.org/abs/1510.08484

[109] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition."

[110] 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," in *Proc. Speech Synthesis Workshop (SSW)*, 2016.

[111] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. INTERSPEECH*, 2020.

[112] Y. Zhang, W. Wang, and P. Zhang, "The effect of silence and dual-band fusion in anti-spoofing system," in *Proc. INTERSPEECH*, 2021.

[113] B. C. Moore, *An introduction to the psychology of hearing.* Brill, 2012.

[114] T. Kinnunen, M. Sahidullah, I. Kukanov, H. Delgado, M. Todisco, A. Sarkar, N. B. Thomsen, V. Hautamäki, N. Evans, and Z. H. Tan, "Utterance verification for text-dependent speaker recognition: a comparative assessment using the RedDots corpus," in *Proc. INTERSPEECH*, San Francisco, USA, 2016, pp. 430–434.

[115] J. Youngberg and S. Boll, "Constant-q signal analysis and synthesis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, Tulsa, Oklahoma, USA, 1978, pp. 375–378.

[116] J. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America (JASA)*, vol. 89, no. 1, pp. 425–434, 1991.

[117] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Proc. Audio Engineering Society International Conference on Semantic Audio*, London, UK, 2014.

[118] G. A. Velasco, N. Holighaus, M. Dorfler, and T. Frill, "Constructing an invertible constant-Q transform with nonstationary Gabor frames," in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, 2011, pp. 93–99.

[119] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 653–656.

[120] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[121] T. Kinnunen, K. Lee, H. Delgado, N. Evans, M. Todisco, J. Sahidullah, M.and Yamagishi, and D. A. Reynolds, "t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Speaker Odyssey Workshop*, Les Sables d'Olonne, France, 2018, pp. 312–319.

[122] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice.* Pearson Education, 2006.

[123] J. R. Deller Jr, J. G. Proakis, and J. H. Hansen, *Discrete time processing of speech signals.* Prentice Hall PTR, 1993.

[124] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah, "Investigation of sub-band discriminative information between spoofed and genuine speech," in *Proc. INTERSPEECH*, 2016, pp. 1710–1714.

[125] J. Brown, "Computer identification of musical instruments using pattern recognition with cepstral coefficients as features," *The Journal of the Acoustical Society of America (JASA)*, vol. 105, no. 3, pp. 1933–1941, 1999.

[126] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification," in *Proc. INTERSPEECH*, Graz, Austria, 2019, pp. 1118–1112.

[127] B. Chettri, T. Kinnunen, and E. Benetos, "Subband modeling for spoofing detection in automatic speaker verification," in *Proc. Speaker Odyssey Workshop*, 2020.

[128] J. Lu, Y. Zhang, W. Wang, and P. Zhang, "Robust cross-subband countermeasure against replay attacks," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 126–132.

[129] J. C. Brown, "Calculation of a constant Q spectral transform," *The Journal of the Acoustical Society of America (JASA)*, vol. 89, no. 1, pp. 425–434, 1991.

[130] S. Umesh and R. Sinha, "A study of filter bank smoothing in mfcc features for recognition of children's speech," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 8, pp. 2418–2430, 2007.

[131] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[132] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman lectures on physics; New millennium ed.* New York, NY: Basic Books, 2010, originally published 1963-1965. [Online]. Available: https://cds.cern.ch/record/1494701

[133] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[134] M. Todisco, H. Delgado, K. A. Lee, M. Sahidullah, N. Evans, T. Kinnunen, and J. Yamagishi, "Integrated presentation attack detection and automatic speaker verification: Common features and gaussian back-end fusion," in *Proc. INTERSPEECH*, 2018.

[135] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[136] H. Muckenhirn, V. Abrol, M. Magimai-Doss, and S. Marcel, "Understanding and visualizing raw waveform-based cnns." in *Proc. INTERSPEECH*, 2019, pp. 2345–2349.

[137] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. INTERSPEECH*, 2015, pp. 1–5.

[138] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[139] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *Proc. ICASSP*, 2018, pp. 5349–5353.

[140] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in *Proc. INTERSPEECH*, 2019, pp. 1268–1272.

[141] J.-h. Kim, H.-j. Shim *et al.*, "RawNeXt: Speaker verification system for variable-duration utterances with deep layer aggregation and extended dynamic scaling policies," in *Proc. ICASSP*, 2022.

[142] H. Lee, Y. Tso, Y. Chang *et al.*, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *Proc. ICASSP*, 2014, pp. 1660–1664.

[143] H. Heo, I. Yang, M. Kim, S. Yoon, and H. Yu, "Advanced b-vector system based deep neural network as classifier for speaker verification," in *Proc. ICASSP*, 2016, pp. 5465–5469.

[144] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. IEEE SLT*, 2018.

[145] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, "Improved rawnet with filter-wise rescaling for text-independent speaker verification using raw waveforms," in *Proc. INTERSPEECH*, 2020, pp. 1496–1500.

[146] H. Dinkel, N. Chen, Y. Qian, and K. Yu, "End-to-end spoofing detection with raw waveform CLDNNS," in *Proc. ICASSP*. IEEE, 2017, pp. 4860–4864.

[147] P. Korshunov, S. Marcel, H. Muckenhirn, A. R. Gonçalves, A. S. Mello, R. V. Violato, F. O. Simoes, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi *et al.*, "Overview of BTAS 2016 speaker anti-spoofing competition," in *Proc. International conference on biometrics theory, applications and systems (BTAS)*. IEEE, 2016, pp. 1–6.

[148] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2015, pp. 1–6.

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

[150] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. ECCV*, 2016.

[151] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[152] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. NIPS*, 2017, pp. 972–981.

[153] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[154] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling. arxiv 2014," *arXiv preprint arXiv:1412.3555*, 2014.

[155] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[156] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[157] H. Tak, J. Patino, A. Nautsch *et al.*, "Spoofing Attack Detection using the Non-linear Fusion of Sub-band Classifiers," in *Proc. INTERSPEECH*, 2020, pp. 1106–1110.

[158] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[159] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[160] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[161] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2009.

[162] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[163] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[164] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[165] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.

[166] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[167] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-shot audio classification with attentional graph neural networks." in *INTERSPEECH*, 2019, pp. 3649–3653.

[168] R. Liu, B. Sisman, and H. Li, "Graphspeech: Syntax-aware graph attention network for neural speech synthesis," in *Proc. ICASSP*, 2021.

[169] J.-w. Jung, H.-S. Heo, H.-J. Yu, and J. S. Chung, "Graph attention networks for speaker verification," in *Proc. ICASSP*, 2021.

[170] P. Tzirakis, A. Kumar, and J. Donley, "Multi-channel speech enhancement using graph neural networks," in *Proc. ICASSP*, 2021, pp. 3415–3419.

[171] Y. Kwon, H.-S. Heo, J.-w. Jung, Y. J. Kim, B.-J. Lee, and J. S. Chung, "Multi-scale speaker embedding-based graph attention networks for speaker diarisation," in *Proc. ICASSP*, 2022, pp. 8367–8371.

[172] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[173] H. Tak, J.-w. Jung, J. Patino, M. Todisco, and N. Evans, "Graph attention networks for anti-spoofing," in *Proc. INTERSPEECH*, 2021.

[174] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019.

[175] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning (ICML)*, 2015.

[176] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *Proc. ICASSP*, 2021.

[177] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning (PMLR)*, 2019, pp. 2083–2092.

[178] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. ICML*, 2019.

[179] G. Xu, W. Li, and J. Liu, "A social emotion classification approach using multi-model fusion," *Future Generation Computer Systems*, vol. 102, pp. 347–356, 2020.

[180] H. Wang, Y. Zou, and W. Wang, "Specaugment++: A hidden space data augmentation method for acoustic scene classification," in *Proc. INTERSPEECH*, 2021.

[181] P. Parasu, J. Epps, K. Sriskandaraja, and G. Suthokumar, "Investigating light-resnet architecture for spoofing detection under mismatched conditions," in *Proc. INTERSPEECH*, 2020, pp. 1111–1115.

[182] G. Hua, A. Beng jin teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Processing Letters*, 2021.

[183] H. Ling, L. Huang, J. Huang, B. Zhang, and P. Li, "Attention-based convolutional neural network for ASV spoofing detection," in *Proc. INTERSPEECH 2021*, 2021, pp. 4289–4293.

[184] W. Ge, J. Patino, M. Todisco, and N. Evans, "Raw differentiable architecture search for speech deepfake and spoofing detection," in *Proc. ASVspoof workshop*, 2021.

[185] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated res2net: Towards robust detection of synthetic speech attacks," in *Proc. INTERSPEECH*, 2021.

[186] X. Wang and J. Yamagishi, "A Comparative Study on Recent Neural Spoofing Countermeasures for Synthetic Speech Detection," in *Proc. INTERSPEECH*, 2021.

[187] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[188] X. Ma, T. Liang, S. Zhang, S. Huang, and L. He, "Improved lightcnn with attention modules for asv spoofing detection," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[189] Y. Ma, Z. Ren, and S. Xu, "RW-Resnet: A novel speech anti-spoofing model using raw waveform," in *Proc. INTERSPEECH 2021*, 2021, pp. 4144–4148.

[190] W. Ge, M. Panariello, J. Patino, M. Todisco, and N. Evans, "Partially-connected differentiable architecture search for deepfake and spoofing detection," in *Proc. INTERSPEECH*, 2021.

[191] Z. Lei, Y. Yang, C. Liu, and J. Ye, "Siamese convolutional neural network using gaussian probability feature for spoofing speech detection," in *Proc. INTERSPEECH*, 2020, pp. 1116–1120.

[192] Z. Wu, R. K. Das, J. Yang, and H. Li, "Light convolutional neural network with feature genuinization for detection of synthetic speech attacks," in *Proc. INTERSPEECH*, 2020.

[193] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[194] 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.

[195] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[196] S. Choi, I.-Y. Kwak, and S. Oh, "Overlapped frequency-distributed network: Frequency-aware voice spoofing countermeasure," in *Proc. INTERSPEECH 2022*, 2022, pp. 3558–3562.

[197] D. Paul, M. Sahidullah *et al.*, "Generalization of spoofing countermeasures: A case study with ASVspoof 2015 and BTAS 2016 corpora," in *Proc. ICASSP*, 2017.

[198] R. K. Das, J. Yang *et al.*, "Assessing the scope of generalized countermeasures for anti-spoofing," in *Proc. ICASSP*, 2020.

[199] J. Yamagishi, X. Wang *et al.*, "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," in *Proc. ASVspoof 2021 Workshop*, 2021.

[200] J. Wang, L. Perez *et al.*, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, vol. 11, 2017.

[201] T. Ko, V. Peddinti *et al.*, "Audio augmentation for speech recognition," in *Proc. INTERSPEECH*, 2015.

[202] T.-L. Vu, Z. Zeng *et al.*, "Audio codec simulation based data augmentation for telephony speech recognition," in *Proc. APSIPA ASC*, 2019.

[203] C. Zhang, S. Ranjan *et al.*, "An analysis of transfer learning for domain mismatched text-independent speaker verification," in *Proc. Odyssey Workshop*, 2018.

[204] A. Krizhevsky, I. Sutskever *et al.*, "ImageNet classification with deep convolutional neural networks," *NeurIPS*, vol. 25, 2012.

[205] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML WDLASL*, 2013.

[206] E. Kharitonov, M. Rivière *et al.*, "Data augmenting contrastive learning of speech representations in the time domain," in *Proc. IEEE SLT*, 2021.

[207] G. Kim, D. K. Han *et al.*, "SpecMix: A mixed sample data augmentation method for training withtime-frequency domain features," in *Proc. INTERSPEECH*, 2021.

[208] Y. Ma, Z. Ren, and S. Xu, "Rw-resnet: A novel speech anti-spoofing model using raw waveform," in *Proc. INTERSPEECH*, 2021.

[209] H. Guo and H. L. Viktor, "Boosting with data generation: Improving the classification of hard to learn examples," in *Proc. IEA/AIE*, 2004.

[210] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint:1510.08484*, 2015.

[211] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[212] A. Y. Kibangou and G. Favier, "Wiener-hammerstein systems modeling using diagonal volterra kernels coefficients," *IEEE signal processing letters*, vol. 13, 2006.

[213] A. V. Oppenheim and R. W. Schafe, "Discrete-time signal processing (3rd Ed.)," 2011.

[214] C.-W. Kok and T. Q. Nguyen, "Multirate filter banks and transform coding gain," *IEEE transactions on signal processing*, vol. 46, 1998.

[215] S. Yin, C. Liu *et al.*, "Noisy training for deep neural networks in speech recognition," *EURASIP JASM*, vol. 2015, 2015.

[216] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. S. Chung, "Augmentation adversarial training for unsupervised speaker recognition," in *Workshop on SAS, NeurIPS*, 2020.

[217] U. Tiwari, M. Soni *et al.*, "Multi-conditioning and data augmentation using generative noise model for speech emotion recognition in noisy conditions," in *Proc. ICASSP*, 2020.

[218] R. Yang, "Additive noise detection and its application to audio forensics," in *Proc. APSIPA*, 2014.

[219] C. Hanilci, T. Kinnunen, M. Sahidullah, and A. Sizov, "Spoofing detection goes noisy: An analysis of synthetic speech detection in the presence of additive noise," *Speech Communication*, vol. 85, pp. 83–97, 2016.

[220] X. Tian, Z. Wu, X. Xiao, E. S. Chng, and H. Li, "An investigation of spoofing speech detection under additive noise and reverberant conditions," in *Proc. INTERSPEECH*, 2016, pp. 1715–1719.

[221] H. Delgado, N. Evans *et al.*, "ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," in *arXiv preprint arXiv:2109.00535*, 2021.

[222] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, 2000.

[223] H. Zhang, M. Cisse *et al.*, "mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[224] X. Wang, X. Qin *et al.*, "The DKU-cmri system for the ASVspoof 2021 challenge: Vocoder based replay channel response estimation," in *Proc. ASVspoof 2021 Workshop (INTERSPEECH satellite)*, 2021.

[225] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," in *Proc. NIPS*, 2018.

[226] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc NIPS*, 2020.

[227] S.-w. Yang, P.-H. Chi *et al.*, "SUPERB: Speech processing universal performance benchmark," in *Proc. INTERSPEECH*, 2021.

[228] A. Conneau, A. Baevski *et al.*, "Unsupervised cross-lingual representation learning for speech recognition," in *Proc. INTERSPEECH*, 2021.

[229] K. Kawakami, L. Wang *et al.*, "Learning robust and multilingual speech representations," in *Proc. EMNLP*, 2020.

[230] D. Cai, W. Wang *et al.*, "An iterative framework for self-supervised deep speaker representation learning," in *Proc. ICASSP*, 2021.

[231] A. Tjandra, D. G. Choudhury *et al.*, "Improved language identification through cross-lingual self-supervised learning," *arXiv preprint arXiv:2107.04082*, 2021.

[232] S. Evain, H. Nguyen *et al.*, "LeBenchmark: A reproducible framework for assessing self-supervised representation learning from speech," in *Proc. INTERSPEECH*, 2021.

[233] S. Chen, Y. Wu, C. Wang, S. Liu, Z. Chen, P. Wang, G. Liu, J. Li, J. Wu, X. Yu *et al.*, "Why does self-supervised learning for speech recognition benefit speaker recognition?" in *Proc. INTERSPEECH*, 2022.

[234] Z. Jiang, H. Zhu, L. Peng, W. Ding, and Y. Ren, "Self-supervised spoofing audio detection scheme," in *Proc. INTERSPEECH*, 2020, pp. 4223–4227.

[235] Y. Xie, Z. Zhang *et al.*, "Siamese network with wav2vec feature for spoofing speech detection," in *Proc. INTERSPEECH*, 2021.

[236] X. Wang and J. Yamagishi, "Investigating self-supervised front ends for speech spoofing countermeasures," in *Proc. The Speaker and Language Recognition Workshop (Speaker Odyssey)*, 2022.

[237] F. Alegre, A. Amehraye *et al.*, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," in *Proc. IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013.

[238] A. v. d. Oord, Y. Li *et al.*, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[239] Y.-A. Chung, Y. Belinkov *et al.*, "Similarity analysis of self-supervised speech representations," in *Proc. ICASSP*, 2021.

[240] S. Schneider, A. Baevski *et al.*, "Wav2vec: Unsupervised pre-training for speech recognition," in *INTERSPEECH*, 2019.

[241] W.-N. Hsu, B. Bolte *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[242] W.-N. Hsu, Y.-H. H. Tsai *et al.*, "HuBERT: How much can a bad teacher benefit ASR pre-training?" in *Proc. ICASSP*, 2021.

[243] Q. Xu, A. Baevski *et al.*, "Self-training and pre-training are complementary for speech recognition," in *Proc. ICASSP*, 2021.

[244] S. Chen, C. Wang *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.

[245] X. Xu, Y. Kang *et al.*, "Explore wav2vec 2.0 for mispronunciation detection," in *Proc. INTERSPEECH*, 2021.

[246] L. Peng, K. Fu *et al.*, "A study on fine-tuning wav2vec2.0 model for the task of mispronunciation detection and diagnosis," in *Proc. INTERSPEECH*, 2021.

[247] N. Vaessen and D. A. van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," *arXiv preprint arXiv:2109.15053*, 2021.

[248] Z. Fan, M. Li *et al.*, "Exploring wav2vec 2.0 on speaker verification and language identification," in *Proc. INTERSPEECH*, 2021.

[249] L. Pepino, P. Riera *et al.*, "Emotion recognition from speech using wav2vec 2.0 embeddings," in *Proc. INTERSPEECH*, 2021.

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

[251] J. M. Martín-Doñas and A. Álvarez, "The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge," in *ICASSP 2022-2022 IEEE Proc. ICASSP*, 2022, pp. 9241–9245.

[252] Y. Eom, Y. Lee, J. S. Um, and H. Kim, "Anti-spoofing using transfer learning with variational information bottleneck," in *Proc. INTERSPEECH*, 2022.

[253] A. Babu, C. Wang, A. Tjandra *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. INTERSPEECH*, 2022, pp. 2278–2282.

[254] M. Ott, S. Edunov *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-HLT 2019: Demonstrations*, 2019.

[255] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. INTERSPEECH*, 2018.

[256] Y. Jung, S. M. Kye *et al.*, "Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances," in *Proc. INTERSPEECH*, 2020.

[257] W. Xie, A. Nagrani *et al.*, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. ICASSP*, 2019.

[258] H.-j. Shim, J. Heo *et al.*, "Graph attentive feature aggregation for text-independent speaker verification," in *Proc. ICASSP*, 2022.

[259] X. Wang and J. Yamagishi, "Investigating active-learning-based training data selection for speech spoofing countermeasure," in *Proc. SLT*, 2022.

[260] ——, "Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders," in *Proc. ICASSP (to appear)*, 2023.

[261] N. M. Müller, F. Dieckmann *et al.*, "Speech is silver, silence is golden: What do ASVspoof-trained models really learn?" in *Proc. ASVspoof 2021 Workshop*, 2021.

[262] D. Mari, F. Latora, and S. Milani, "The sound of silence: Efficiency of first digit features in synthetic audio detection," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2022, pp. 1–6.

[263] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2019, pp. 1–10.

[264] Z. Zhang, Y. Gu, X. Yi, and X. Zhao, "Synspeechddb: a new synthetic speech detection database," 2020.

[265] A. Gomez-Alanis, J. A. Gonzalez-Lopez, and A. M. Peinado, "GANBA: Generative adversarial network for biometric anti-spoofing," *Applied Sciences*, no. 3, 2022.

[266] H. Wu, P.-C. Hsu, J. Gao, S. Zhang, S. Huang, J. Kang, Z. Wu, H. Meng, and H.-Y. Lee, "Adversarial sample detection for speaker verification by neural vocoders," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 236–240.

[267] J. Villalba, Y. Zhang, and N. Dehak, "x-Vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification." in *INTERSPEECH*, 2020, pp. 4233–4237.

[268] S. Joshi, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4811–4826, 2021.

[269] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8261–8265.

[270] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting deep-fake videos from phoneme-viseme mismatches," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 660–661.

[271] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5039–5049.