Audio Spoofing Detection via Hybrid Feature Integration

A THESIS

submitted by

BARNEET SINGH

MT23028

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

Department of Computer Science and Engineering

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

May 2025

THESIS CERTIFICATE

This is to certify that the thesis titled **Audio Spoofing Detection via Hybrid Feature Integration**, submitted by **Barneet Singh**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **Master of Technology**, in Computer Science Engineering Specialization in Artificial Intelligence, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Vinayak Abrol
Thesis Supervisor
Assistant Professor
Dept. of Computer Science Engineering
IIIT Delhi, 110020

Place: New Delhi
Date: May 21, 2025

ACKNOWLEDGEMENTS

I am sincerely grateful to the Infosys Center for AI (CAI) at IIIT Delhi for providing the necessary computational resources and academic environment that facilitated this research. Their unwavering support has been instrumental in the successful completion of this work.

I am sincerely grateful to Dr. Vinayak Abrol, whose exceptional depth of knowledge in audio processing and computational linguistics has not only shaped the direction of this work but has also served as a constant source of inspiration and learning. His unwavering dedication to research has been an immense source of inspiration. His ability to break down complex concepts and continuous encouragement to think critically have significantly shaped my understanding of language translation systems.

I am also grateful to the members of the Cross-Caps Laboratory, whose discussions and insights enriched the development of this project. Their willingness to share knowledge, engage in critical problem-solving sessions, and provide constructive feedback fostered a collaborative environment that significantly enhanced thisresearch'ss technical and conceptual aspects. Special thanks to those who took the time to review my work, challenge my assumptions, and contribute valuable suggestions that shaped the outcomes.

Furthermore, I appreciate my friends and colleagues for their endless support and motivation, particularly during challenging phases of balancing coursework and thesis work. Their encouragement kept me grounded and focused throughout this rigorous process.

Lastly, my heartfelt thanks to my family, whose constant belief in my potential has been my most significant source of strength. Their steadfast support and patience have been fundamental to the successful completion of this thesis.

I am sincerely grateful to all for the invaluable support and unwavering confidence in my work.

Barneet Singh

ABSTRACT

KEYWORDS: Anti-Spoofing; Deepfake Detection; Speaker Verification; ASVspoof Challenge; Wav2Vec2; WavLM; UniSpeech; ECAPA-TDNN; Hybrid Feature Integration; Self-Supervised Learning

This thesis explores advanced techniques in the field of audio spoofing detection. With the emergence of high-quality deepfake generation techniques and the vulnerabilities in automatic speaker verification (ASV) systems, robust countermeasures are essential. We investigate state-of-the-art deep learning models including ECAPA-TDNN, ResNet, TitaNet, and self-supervised models such as Wav2Vec2, WavLM, and UniSpeech. Experiments are conducted on datasets from ASVspoof 2021 and 2024 challenges. Our approach introduces a hybrid integration of handcrafted features with SSL-based embeddings, demonstrating notable improvements in Equal Error Rate (EER) and minimum Detection Cost Function (minDCF). Data augmentation strategies are also evaluated for enhancing robustness. Results indicate that hybrid systems combining engineered and learned features outperform standalone models and offer practical insights for developing next-generation anti-spoofing solutions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT ii

LIST OF TABLES v

LIST OF FIGURES vi

ABBREVIATIONS vii

NOTATION viii

1 INTRODUCTION 1

1.1 Motivation and Research Gap 4

1.2 Objectives 4

1.3 Research Questions 5

2 Literature Review 7

2.1 Spoofing Attacks and ASVspoof Challenges 7

2.1.1 Types of Spoofing Attacks: 7

2.1.2 ASVspoof Challenge Evolution: 7

2.2 Anti-Spoofing Features and Techniques 9

2.2.1 Handcrafted Features 9

2.2.2 Traditional Classifiers 10

2.2.3 CNN and ResNet Models 10

2.2.4 Time-Delay Neural Networks (TDNNs) and ECAPA-TDNN 11

2.2.5 Graph-Based Models (AASIST) 11

2.2.6 Transformers and Self-Supervised Models 12

2.2.7 Data Augmentation and Training Strategies 13

3 Methodology 15

3.1 System Overview 15
3.2 Baseline Model Architectures 17
3.3 Self-Supervised Pre-Trained Models 19
3.4 Hybrid Feature Integration Approach 21
3.5 Experimental Pipeline 23

4 Experimentation **27**
4.1 Datasets and Preparation 27
4.2 Experiments Conducted 29
4.3 Implementation Details and Reproducibility 31

5 Results and Analysis **33**
5.1 Baseline Model Performance on ASVspoof 2021 DF 33
5.2 Self-Supervised Models on ASVspoof 2024 35
5.3 Hybrid Feature Integration Results 37
5.4 Evaluation on ASVspoof 2024 Evaluation Set 40
5.5 Discussion of Key Findings 42

6 Discussion **45**
6.1 Addressing the Research Questions 45
6.2 Comparison with Prior Work 48
6.3 Practical Implications 49
6.4 Future Work 50
6.5 Summary of Contributions 51

7 CONCLUSION **52**
7.1 Summary of Findings 52
7.2 Implications for Anti-Spoofing Research 53
7.3 Conclusion and Future Outlook 54

LIST OF TABLES

5.1 Performance of SSL Models on ASVspoof 2024 Dev Set (EER in %, minDCF in parentheses) 35

LIST OF FIGURES

1.1 Type of spoofing attacks 1

2.1 ASVspoof Challenge Evolution 8

2.2 AASIST Model framework 12

3.1 Deep Learning Model pipeline 16

3.2 SSL Model pipeline 16

3.3 Hybrid Feature pipeline 16

4.1 ASVspoof 2024 Dataset 28

5.1 Equal Error Rates of baseline models (ECAPA-TDNN, ResNet-18, TitaNet) on ASVspoof 2021 deepfake development set. Lower EER is better; ResNet achieved the lowest EER (best performance) among the three. 34

5.2 Impact of data augmentation on EER (%) for WavLM, Wav2Vec2, UniSpeech models on ASVspoof 2024 dev. For each model, the orange bar is without augmentation and the red bar is with augmentation. Augmentation significantly lowers EERs for all models. 36

5.3 EER (%) of WavLM with different integrated features on ASVspoof 2024 dev set. PLP, MFCC, LPC significantly improved performance over WavLM alone, while LFCC and RASTA features degraded it. Note that WavLM alone (from previous results) was 5.0% EER for reference. 38

ABBREVIATIONS

Abbreviation	Description
EER	Equal Error Rate
DCF	Detection Cost Function
minDCF	Minimum Detection Cost Function
AASIST	Audio Anti-Spoofing using Integrated Spectro-Temporal features
PLP	Perceptual Linear Prediction
RASTA	Relative Spectral Transform - Perceptual Linear Prediction
MFCC	Mel-Frequency Cepstral Coefficients
CM	Counter Measures
AI	Artificial Intelligence
TTS	Text-to-Speech
GMM	Gaussian Mixture Model
ASV	Automatic Speaker Verification
ECAPA-TDNN	Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Network
SASV	Spoof-Aware Speaker Verification
SSL	Self-Supervised Learning
LPC	Linear Predictive Coding
LFCC	Linear Frequency Cepstral Coefficients

NOTATION

$P_{fa}(\theta)$	False alarm rate at threshold θ
$P_{miss}(\theta)$	Miss (false rejection) rate at threshold θ
EER	Equal Error Rate: $P_{fa}(\theta_{EER}) = P_{miss}(\theta_{EER})$
τ_{cm}	Detection threshold for countermeasure system
$P^{cm}_{miss}(\tau_{cm})$	Miss rate for bonafide trials at threshold τ_{cm}
$P^{cm}_{fa}(\tau_{cm})$	False alarm rate for spoofed trials at threshold τ_{cm}
DCF	Detection Cost Function: weighted sum of P^{cm}_{miss} and P^{cm}_{fa}
C_{miss}	Cost of miss (rejecting bonafide as spoof)
C_{fa}	Cost of false alarm (accepting spoof as bonafide)
π_{spf}	Prior probability of spoofed trials
β	Cost scaling factor: $\frac{C_{miss}}{C_{fa}} \cdot \frac{1-\pi_{spf}}{\pi_{spf}}$
minDCF	Minimum Detection Cost Function over τ_{cm}
DCF'(τ_{cm})	Normalized DCF: $\beta \cdot P^{cm}_{miss}(\tau_{cm}) + P^{cm}_{fa}(\tau_{cm})$
$\mathbf{DCF}_{def}$	Default cost: $\min\{C_{miss} \cdot (1-\pi_{spf}), C_{fa} \cdot \pi_{spf}\}$

CHAPTER 1

INTRODUCTION

Automatic speaker verification (ASV) systems offer a convenient biometric solution for voice-based authentication. However, it has been well established that ASV systems are vulnerable to various spoofing attacks. Attackers can manipulate or synthesize speech to impersonate a target speaker, thereby undermining the security of voice authentication. Spoofing attacks can take multiple forms, including human mimicry (impersonation), text-to-speech (TTS) synthesis, voice conversion (VC), and replay attacks, as well as modern AI-based deepfake audio generation. The ASVspoof initiative was launched to systematically tackle this problem by providing public challenge datasets and metrics, spurring development of anti-spoofing countermeasures (CMs) that distinguish bona fide (genuine) speech from spoofed speech.

Over the past decade, ASVspoof challenges have evolved to address increasingly sophisticated attacks. Early editions (2015, 2017) focused on limited attack types (speech synthesis, voice conversion, replay), whereas later editions introduced more diverse and unpredictable attacks. The ASVspoof 2019 challenge included logical access (LA, TTS/VC attacks) and physical access (PA, replay attacks) scenarios. Notably, results from ASVspoof 2019 highlighted the challenge of generalization: systems that performed well on known attack algorithms often faltered against unseen attacks. For

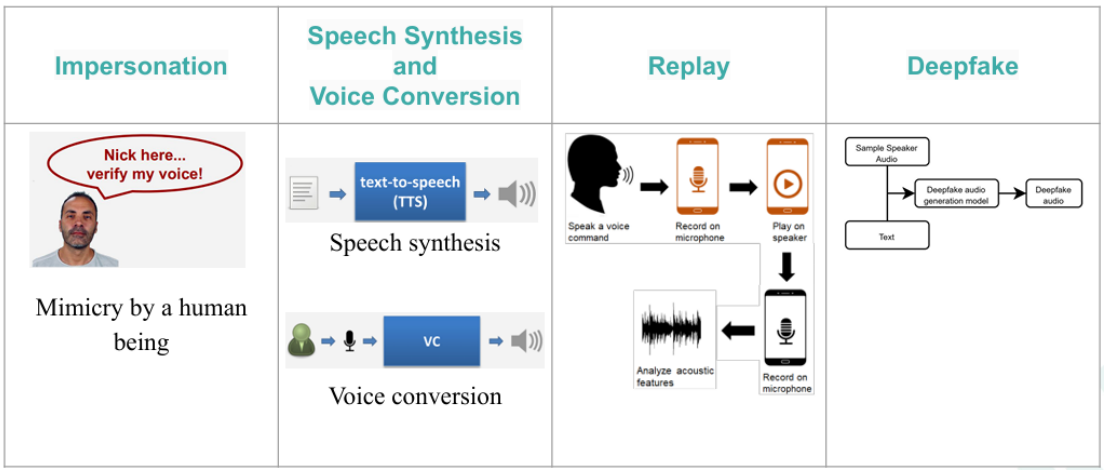

Figure 1.1: Type of spoofing attacks

instance, the best baseline system's EER (equal error rate) on ASVspoof 2021's logical access task jumped from 0.55% on the development set to 9.26% on the evaluation set, underscoring the "evaluation gap" problem wherein new, unforeseen attack methods can dramatically degrade detection performance. This motivates research into more robust models that can handle a wide range of spoofing techniques encountered "in the wild."

Deepfake audio has emerged as a particularly alarming threat in recent years. Deepfakes involve using advanced TTS and VC algorithms (often powered by deep neural networks) to produce synthetic speech that closely mimics a target speaker's voice. The ASVspoof 2021 challenge introduced a Deepfake (DF) track separate from LA and PA, reflecting growing concern over AI-generated voice spoofing. The DF dataset greatly expanded the variety of attacks, including many more TTS/VC algorithms (over 100 in evaluation) and even domain-mismatched attacks. Meanwhile, the latest challenge, ASVspoof 5 (2024), further raises the bar by using a crowdsourced, large-scale dataset with vastly more speakers and acoustic diversity, plus the introduction of adversarial attacks combined with spoofing for the first time. The ASVspoof 5 dataset is derived from the Multilingual Librispeech corpus (English portion) and contains no overlap of speakers or attack algorithms between training, development, and evaluation partitions, ensuring rigorous generalization testing. It is also gender-balanced and includes compression artifacts (both traditional codecs like MP3 and neural codecs) to simulate real-world conditions. These developments highlight that modern anti-spoofing systems must handle a broad spectrum of attack types and acoustic conditions, from high-quality deepfakes to adversarially perturbed audio.

State-of-the-art countermeasures for spoofing detection have correspondingly advanced alongside the threat. Early systems relied on hand-crafted acoustic features (e.g., MFCC, CQCC, LFCC) fed into classifiers like GMMs or SVMs. In recent years, researchers have increasingly applied deep learning, yielding significant performance gains. Numerous architectures have been explored: convolutional neural networks (CNNs) (including light CNNs with max-feature-map activations), deep ResNet variants, time-delay neural networks (TDNNs) such as the x-vector and enhanced TDNN models, and Transformers. Notably, ResNet-based models have been popular in challenge submissions; for example, the winning systems in ASVspoof 2019 and 2021 leveraged ResNet backbones with spectrogram or cepstral features, achieving strong

spoof detection performance. Another line of work introduced RawNet – end-to-end models operating on raw waveforms – and their successors. One cutting-edge approach, AASIST (Audio Anti-Spoofing using Integrated Spectro-Temporal graph), combined a RawNet2 front-end with graph attention networks to jointly model temporal and spectral artifact patterns. AASIST demonstrated a 20% relative improvement over the prior state-of-the-art, without needing an ensemble. Subsequent research built on AASIST's success; for instance, enhancements like AASIST2 and AASIST3 introduced additional modules and utilized self-supervised features to further improve generalization. The recently proposed AASIST3 model (for the ASVspoof 2024 challenge) incorporates Kolmogorov–Arnold networks (KAN) into AASIST and other refinements, achieving more than two-fold performance improvement over the original (minDCF 0.5357 $\rightarrow$ 0.1414 under certain conditions). These results indicate that combining powerful front-ends with sophisticated back-ends can markedly boost deepfake detection accuracy.

A particularly promising direction is the use of self-supervised learning (SSL) based models for anti-spoofing. Models like Wav2Vec 2.0, WavLM, and UniSpeech are pre-trained on massive audio corpora with self-supervised objectives, enabling them to learn rich general-purpose speech representations. Researchers have begun fine-tuning such models for spoof detection, with impressive outcomes. Tak et al. (2022) showed that a fine-tuned Wav2Vec2 front-end, when combined with data augmentation and a simple classifier, yielded record-low EERs on ASVspoof 2021 (LA EER 1.19%, DF EER 4.38%) – a relative EER reduction of up to 90% compared to a traditional sinc-filter front-end. This underscores the benefit of leveraging large-scale pre-training to improve robustness against diverse attacks. However, these SSL models are typically very large (e.g., hundreds of millions of parameters). As noted by Tak et al., model complexity and practicality become concerns: Wav2Vec2 is "massively more complex" than prior solutions, and it remains unclear if such large models can be deployed in real-world, resource-constrained settings. Thus, there is a dual challenge: to push detection performance higher (especially for unseen attacks) while also ensuring solutions remain efficient and usable in production.

1.1 Motivation and Research Gap

Given the above landscape, this thesis is motivated by the need for robust, generalizable, yet efficient spoofing detection techniques. Despite progress, gaps remain. Traditional single-stream models may not fully exploit complementary information present in different feature representations of audio. Meanwhile, purely learned features from SSL models, though powerful, might overlook some subtle spoofing cues that carefully engineered acoustic features can capture. We hypothesize that a hybrid feature integration approach – combining deep learned embeddings with handcrafted features that can leverage the strengths of both. Prior studies have hinted at this: for example, Wang et al. fused multiple audio representations to improve an AASIST-based system's generalization. However, the optimal way to integrate features and the extent of gains achievable were open questions. Furthermore, evaluating such hybrid systems on the latest challenging datasets (like ASVspoof 2024) can illuminate their effectiveness against state-of-the-art attacks. In summary, the key research gaps addressed are: (1) assessing the performance of advanced speaker recognition models (ECAPA-TDNN, ResNet, TitaNet) on deepfake audio detection, (2) determining the effectiveness of SSL pre-trained models (Wav2Vec2, WavLM, UniSpeech) for spoofing detection, and (3) exploring whether integrating classic speech features with SSL embeddings yields improved detection accuracy, thereby offering a route to robust performance without solely relying on extremely large models.

1.2 Objectives

Based on these gaps, the objectives of this research are defined as follows:

- Evaluate the effectiveness of three baseline deep models – ECAPA-TDNN, ResNet, and TitaNet – for detecting audio deepfakes using the ASVspoof 2021 DF dataset. This will establish a performance baseline of conventional speaker recognition architectures on spoofing detection.
- Investigate the performance of leading self-supervised speech models (Wav2Vec2, WavLM, UniSpeech) on the ASVspoof 2024 deepfake dataset. We aim to quantify the improvement these SSL models can provide over baseline systems and identify which model is most suitable for this task.
- Determine the impact of data augmentation strategies on spoofing detection performance for SSL-based models. Given prior evidence of augmentation aiding

generalization, we incorporate augmentation during training and compare results with and without it.

- Propose and evaluate a hybrid feature integration approach that combines SSL model embeddings with handcrafted acoustic features (such as PLP, MFCC, etc.). The goal is to assess whether this fusion can further reduce error rates by capturing complementary spoofing evidence.
- Analyze the results to derive insights into the strengths and limitations of each approach. We specifically seek to identify which features or model components contribute most to spoof detection, the remaining weaknesses (e.g., certain attack types that are still challenging), and how the research findings can guide development of more secure ASV systems.

1.3 Research Questions

In line with the above objectives, the thesis addresses the following research questions:

- How do conventional deep speaker recognition models (ECAPA-TDNN, ResNet, TitaNet) perform in detecting spoofed (deepfake) speech? This explores whether architectures tuned for speaker identity can effectively detect fake vs. real speech, and which architecture is most suitable.
- Can self-supervised learning models (Wav2Vec2, WavLM, UniSpeech) significantly improve spoofing detection accuracy on a challenging dataset like ASVspoof 2024, compared to traditional models? This asks if SSL pre-training yields a tangible advantage in generalization to new spoof attacks.
- What is the role of data augmentation in enhancing the robustness of deepfake speech detectors? We examine how adding noisy or perturbed examples during training affects the model's ability to handle unforeseen conditions or attacks.
- Does the integration of handcrafted audio features with deep learning embeddings boost detection performance beyond using either alone? This addresses our core hypothesis of hybrid feature integration, investigating if such fusion captures richer indicators of spoofing.
- What are the limitations of the proposed approaches, and how can they be mitigated in future work? This question prompts an analysis of factors like model complexity, computational requirements, performance on evaluation (unseen) data, and how to balance detection accuracy with efficiency.

By answering these questions, this thesis aims to advance the state-of-the-art in audio spoofing detection. The central hypothesis is that a hybrid integration of complementary features can yield a more robust anti-spoofing system than relying on a single

type of representation. Through a series of experiments on recent ASVspoof benchmarks, we seek to validate this hypothesis and contribute new insights into designing countermeasures that safeguard ASV systems from ever-evolving spoofing attacks.

CHAPTER 2

Literature Review

In this chapter, we review prior work on spoofing detection and the various techniques that have been developed to counter voice spoofing. We begin by summarizing the progression of the ASVspoof challenges and the nature of spoofing attacks addressed. Then, we discuss major categories of anti-spoofing methods, including both feature-based approaches and deep learning models, as well as recent state-of-the-art systems. We also cover the incorporation of data augmentation and the emerging use of self-supervised models. This review establishes the context for our research and highlights the motivations for a hybrid feature approach.

2.1 Spoofing Attacks and ASVspoof Challenges

2.1.1 Types of Spoofing Attacks:

Spoofing attacks on speaker verification can be broadly categorized as: (i) human impersonation, where an individual mimics the target speaker's voice; (ii) text-to-speech (TTS) synthesis, where an attacker uses an algorithm to generate speech from text in the target's voice; (iii) voice conversion (VC), where an attacker transforms their voice or another source speaker's voice to sound like the target; and (iv) replay attacks, where a recording of the target's speech is played back to the ASV system. Early studies primarily considered replay and synthetic speech (TTS/VC) attacks in isolation. Impersonation by humans, while a threat, is not easily reproducible or standardized, so it has not been the main focus of ASVspoof databases. Instead, the challenges emphasize attacks generated by algorithms, which can be systematically studied.

2.1.2 ASVspoof Challenge Evolution:

The ASVspoof challenge series (2015–2024) has been the cornerstone of anti-spoofing research. Each edition introduced new facets:

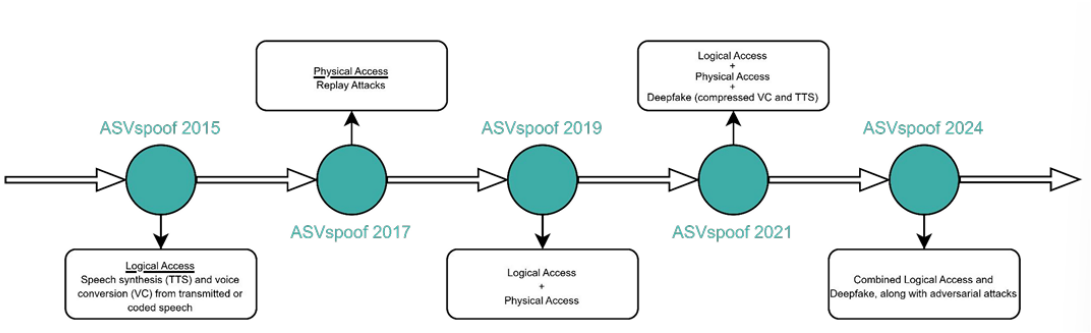

Figure 2.1: ASVspoof Challenge Evolution

- ASVspoof 2015: Focused on logical access attacks using voice conversion and speech synthesis. It provided the first public dataset of bona fide and spoofed utterances, fostering initial research on features like MFCC and phase-based cues for spoof detection.

- ASVspoof 2017: Dedicated to physical access (replay) attacks. This edition collected replayed speech under various room and microphone conditions. The best systems here often used high-frequency details and acoustic scene analysis to detect replay artefacts (e.g., using CQCC features and CNNs).

- ASVspoof 2019: Combined both LA and PA tasks in one challenge. The LA dataset (for TTS/VC attacks) included multiple new synthesis algorithms (total 6 attack types in training) and emphasized generalization by having different attacks in eval. The PA dataset encompassed replay with controlled acoustic simulation. Baseline features were LFCC (linear frequency cepstral coefficients) with GMM, but top systems quickly moved to deep networks. Notably, a Montreal team's system using Squeeze-Excitation ResNet and spectrogram features, and an STC (Speech Technology Center) team's system using ResNet with Max-Feature-Map activation (a variant of Light CNN) were among the top performers. This signaled a shift towards CNN-based learned feature approaches. The pooled EERs in 2019 dropped to just a few percent for known attacks, but remained higher for unseen ones, highlighting generalization issues.

- ASVspoof 2021: Introduced a new Deepfake (DF) track in addition to LA and PA. The deepfake task specifically targeted high-quality TTS/VC attacks not seen in training. The 2021 LA set reused 2019 LA but applied codec compression instead of pure telephony simulation (reflecting modern VoIP scenarios). The DF dataset was created by generating spoofed utterances with many different TTS/VC methods and even included some with minor adversarial perturbations. As mentioned, the gap between dev and eval performance in 2021 DF was stark: the baseline LFCC-LCNN that got 0.5% EER on dev shot up to 9% EER on eval due to unseen attack algorithms. Top challenge submissions in 2021 employed advanced architectures such as RawNet2-based models, ensembles of ResNets, and preliminary use of transformer networks or graph networks. For example, the winning system for DF (by Deepspectrum team) fused multiple models including an EfficientNet and a RawNet-based model. Also, AASIST was introduced around this time (published 2022) and became a new baseline for ASVspoof due to its strong performance.

- ASVspoof 2024 (ASVspoof 5): This latest edition created a single combined

LA+DF task to reflect more realistic conditions where both traditional synthetic speech and modern deepfakes are possible. The dataset is significantly larger and derived from crowdsourced real data: over 4,000 speakers from MLS (thus far more diverse than previous datasets). Spoofing attacks were also crowdsourced – community contributors built TTS/VC models or used various tools to generate attacks, yielding a wide variety. Crucially, adversarial attacks were introduced, meaning some spoofed samples have been intentionally crafted to evade detection by certain surrogate models. This addition acknowledges the growing threat of attackers who actively try to fool anti-spoofing systems (not just ASV). ASVspoof 5 also defined two tracks: Stand-alone CM (what we address) and SASV (spoofing-aware speaker verification) which combines ASV and CM decisions. New evaluation metrics were adopted, especially minDCF (minimum detection cost function) as the primary metric for CM, alongside traditional EER. Baseline results reported for ASVspoof 5 indicated that attacks significantly increased EERs of baseline systems (the baseline CM EER was on the order of 27% for track 1), but top submissions dramatically improved upon this. The evolution from 2015 to 2024 clearly shows an escalation in attack realism and difficulty, necessitating more resilient detection methods.

2.2 Anti-Spoofing Features and Techniques

2.2.1 Handcrafted Features

Early anti-spoofing research focused on identifying distinguishing artifacts in spoofed audio via hand-engineered features. Commonly used features included cepstral coefficients like MFCC, LFCC, and CQCC (Constant-Q Cepstral Coefficients). These features capture spectral envelope information and were favored for their success in speech/speaker recognition. For instance, LFCC was the default feature in ASVspoof 2019 baseline. Such features can pick up subtle differences: e.g., replay attacks often introduce specific high-frequency drop-offs or reverberation patterns, which cepstral features can detect; speech synthesis attacks may exhibit flatter pitch or phase irregularities. Other features explored include spectral flux, group delay, phase-based features (since many TTS systems historically struggled to model phase naturally), and various voice quality features (jitter, shimmer) to catch synthetic glitches. Handcrafted features provided initial traction – for example, a baseline GMM on CQCC achieved respectable performance in ASVspoof 2015. However, the limitation is that they capture a limited set of assumptions; as attack algorithms improved (e.g., neural vocoders producing more natural spectra), the discriminatory power of fixed features diminished.

This spurred the transition to data-driven feature learning.

2.2.2 Traditional Classifiers

Alongside features, early countermeasures used standard classification methods: Gaussian Mixture Models (GMMs) were used in 2015/2017 baselines to model bona fide vs spoof distributions; Support Vector Machines (SVMs) and decision trees were also tried. With the advent of larger datasets (2019 onward), these shallow models struggled to scale and learn complex boundaries in high-dimensional feature space. This opened the door for neural networks that could jointly learn feature representations and classification.

2.2.3 CNN and ResNet Models

Convolutional neural networks quickly became popular for spoofing detection around 2019. CNNs can learn spectral-temporal patterns indicative of spoofing. For example, one hallmark of many VC/TTS attacks is the presence of unnatural smoothing or discontinuities in the spectrogram; CNN filters can potentially pick up on these. A notable architecture was the Light CNN (LCNN) with max-feature-map activation, introduced by Lavrentyeva et al. in ASVspoof 2019. LCNN could learn filters that detect subtle artefacts while the max-out mechanism helped suppress irrelevant features, yielding good performance on replay and synthesis detection. ResNets, with their deeper architecture and skip connections, soon surpassed plain CNNs. Several teams applied ResNet-18/34 or custom ResNet variants to spectrogram or LFCC inputs. These models achieved high accuracy by learning discriminative features—e.g., detecting slight high-frequency noise or vocoder-related patterns that differ between human and machine speech. The ASVspoof 2021 baseline for LA/DF was actually a ResNet (SE-ResNet18) model combined with an attentive statistic pooling layer for utterance-level classification. ResNets remain a strong foundation; indeed, our work includes a ResNet model as one baseline. One challenge with CNN/ResNet approaches is that they can overfit to known spoofing artifacts and sometimes lack generalization to novel attacks unless properly regularized or augmented.

2.2.4 Time-Delay Neural Networks (TDNNs) and ECAPA-TDNN

TDNNs, famously used in x-vector speaker embeddings, were also applied to spoof detection. A TDNN can be seen as 1-D convolution over time with context window; it's adept at capturing temporal patterns. The x-vector approach was tried for anti-spoofing (treating spoof vs bona fide as classes akin to speaker classes). Results were moderate initially. However, an enhanced TDNN called ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation) brought TDNN-based models back to prominence in speaker recognition, and by extension is relevant to anti-spoofing. ECAPA-TDNN introduced Squeeze-Excitation blocks, multi-layer feature aggregation, and an attentive statistics pooling, significantly improving representation power. While ECAPA was designed for speaker discrimation, its robust embeddings might capture subtle cues of audio manipulations too. Indeed, ECAPA-TDNN embeddings have been used as inputs to spoof detectors or as backbones in some systems. In our experiments, we explicitly test ECAPA-TDNN's capability in detecting deepfakes, which to date was not well documented in literature.

2.2.5 Graph-Based Models (AASIST)

A recent innovation was using Graph Neural Networks to handle the fact that spoof artefacts could lie in either spectral or temporal domains or a combination. AASIST (Jung et al., 2022) set a new state-of-art by constructing a bipartite graph of subbands and time frames, then using a Heterogeneous Graph Attention Layer to learn interactions between spectral and temporal nodes. In essence, AASIST's architecture learns to emphasize the domain (time vs frequency) where spoof evidence is stronger for a given input. For example, if a particular synthetic attack has unnatural spectral texture but temporal patterns similar to real speech, the model can focus on spectral nodes; for another attack with weird prosody, it can focus on temporal anomalies. By integrating these via a specialized max graph operation and readout, AASIST achieved superior performance across attacks without needing an ensemble. Its lightweight variant even showed that an efficient model (85k parameters) could outperform heavier models. AASIST's success has spurred follow-ups: AASIST2 introduced modifications to better handle short utterances and integrated Wav2Vec2 features, and AASIST3 (Borodin

et al. 2024) as discussed, added KAN layers and other regularization to further halve the minDCF. These graph-based approaches illustrate the field's move toward domain-agnostic modeling – not assuming a fixed feature space, but letting the model figure out where the telltale artifacts lie.

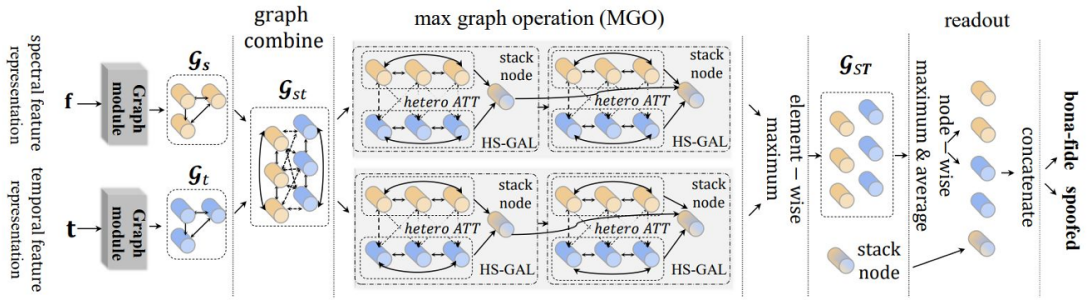

Figure 2.2: AASIST Model framework

2.2.6 Transformers and Self-Supervised Models

Transformer networks, with their self-attention mechanism, have become dominant in many speech tasks. For spoofing detection, researchers have attempted to use Transformers to capture long-range dependencies in audio that might indicate spoofing (e.g., inconsistencies between phoneme segments). One approach was to use Transformers directly on spectrogram sequences or on handcrafted feature sequences. However, training transformers from scratch for anti-spoofing is challenging due to data limitations (spoof datasets, though large by some standards, are still not in the order of magnitude of what's needed to train a transformer from scratch effectively). This is where self-supervised learning (SSL) models enter. SSL models like Wav2Vec 2.0 (Baevski et al. 2020), WavLM (Chen et al. 2022), and UniSpeech (Microsoft, 2021) are pre-trained on tens of thousands of hours of unlabeled audio to learn general speech representations. Their architectures are typically Transformer encoders that output a contextualized representation for each frame or segment of audio. These models have revolutionized speech processing by providing powerful features that can be fine-tuned for specific tasks with relatively smaller labeled datasets. In anti-spoofing, SSL models address the data scarcity problem and provide robust features that might generalize better to unseen conditions. The Odyssey 2022 study by Tak et al. (mentioned earlier) fine-tuned a Wav2Vec2 on ASVspoof data and coupled it with a simple attention pooling + classifier, achieving unprecedented low error rates. Another work by Wang et al. (2022)

found Wav2Vec features, when fused with an AASIST back-end, boosted performance in ASVspoof 2021. WavLM, an improved SSL model that includes speaker-aware pre-training and data augmentation during training, has shown even better performance on speaker tasks and ASR; it is thus a prime candidate for spoof detection. UniSpeech is another SSL model that unifies speech representation learning for ASR and speaker recognition, which could also be beneficial in detecting spoofs by leveraging learned speaker characteristics (e.g., a spoof might not maintain consistent speaker traits). We will review these in context during our experiments. It is important to note a trade-off with SSL models: they are heavy and inference is slower, but they dramatically reduce EERs. A practical system might use these as front-ends but would need optimization for deployment.

2.2.7 Data Augmentation and Training Strategies

Across the literature, one consistently effective technique to improve generalization is data augmentation. Augmentation can simulate variations that the model should learn to handle, thereby preventing overfitting to specific artifact "signatures". For anti-spoofing, common augmentations include adding background noise (from MUSAN dataset) at various SNRs, adding reverberation using room impulse responses, applying band-pass or low-pass filters (to simulate codec compression or bandwidth limitations), time-stretch or pitch-shift, etc. In the context of ASVspoof, augmentation can help models not latch onto trivial cues that won't generalize. For example, if all spoof training data is clean but real data has noise, a model might mistakenly use "presence of noise" as a cue for bonafide; augmenting both classes with noise removes this cue. Tak et al. (2022) systematically explored augmentation and found it complementary to SSL fine-tuning, yielding additional 88–90% relative EER reduction when combined. Similarly, many challenge submissions include augmentation as a standard step. Another strategy is regularization and loss engineering: use of angular margin losses (e.g., AAM-Softmax as in speaker recognition) to make embeddings of bona fide vs spoof more separable, use of one-class learning objectives, or multi-task learning (e.g., training to predict which attack type it is in addition to bona fide/spoof labels). The AASIST3 paper lists various training tricks (SAM – Sharpness-Aware Minimization, ASAM, stochastic weight averaging, etc.) employed to squeeze out better generaliza-

tion. While our work primarily focuses on architecture and features, we also leverage augmentation given its proven benefits.

In summary, the literature shows a clear trajectory: from expert-designed features and simple classifiers to end-to-end deep networks and large pre-trained models. Each generation of methods has reduced spoof detection error rates, yet the core challenge of generalizing to unseen attacks persists. It is also evident that no single feature or model has solved the problem outright – different approaches have different strengths. This motivates research into hybrid methods that can combine complementary strengths. For instance, CNNs are good at local pattern detection, whereas RNNs/Transformers capture global consistency; similarly, handcrafted features might highlight known weaknesses of synthesis (e.g., lack of formant detail) while SSL features provide a broad coverage of speech characteristics. The idea of fusing multiple representations has been hinted at in prior works, but there is room to systematically evaluate such integration on the newest datasets. Our work positions itself in this context: building on the best practices identified (like using SSL features and augmentation) and innovating by integrating classic features with deep embeddings, aiming to advance anti-spoofing toward more resilient performance against the ever-expanding universe of voice spoofing attacks.

CHAPTER 3

Methodology

This chapter details the methodologies and models employed in our study. We describe the architectures of the chosen baseline models (ECAPA-TDNN, ResNet, TitaNet) and the self-supervised models (Wav2Vec2, WavLM, UniSpeech), as well as our proposed approach for hybrid feature integration. We then outline the experimental pipeline, including data preprocessing, training procedures, and evaluation metrics. The goal is to provide a clear understanding of how each model is implemented and how the experiments are designed to answer our research questions

3.1 System Overview

Our overall system for spoofing detection follows a supervised binary classification paradigm: given an input audio utterance, the system outputs a score or decision indicating whether the speech is bona fide (real) or spoofed (fake). The key differentiator between the systems we evaluate is how the feature representation of the audio is obtained and processed.

1. **Front-End Feature Extraction:** This stage transforms the raw audio waveform into a suitable feature representation. Depending on the system, this could be a set of hand-engineered features (e.g., MFCCs), learned embeddings from a pre-trained model (e.g., WavLM), or a combination of both. In some end-to-end models like RawNet2 or ECAPA-TDNN, the front-end is effectively part of the neural network itself (learned filters).
2. **Back-End Classification Model:** The back-end takes the features from the front-end and performs classification. This could be a simple feed-forward network, a graph network (as in AASIST), or a deeper CNN/ResNet. Often, an aggregation layer (such as attentive statistic pooling or a recurrent layer) is used to convert frame-level features to an utterance-level representation, followed by a fully connected layer and a softmax or sigmoid output for the two classes (bonafide vs spoof).

We implemented multiple systems within this framework to compare their performance. Broadly, they can be grouped into three categories:

1. **Speaker-Model Baselines (Trained from Scratch):** ECAPA-TDNN, ResNet, and TitaNet, each trained on spoof data from scratch (or with random initialization). These models represent strong architectures from speaker recognition or general audio classification that we apply to spoof detection.

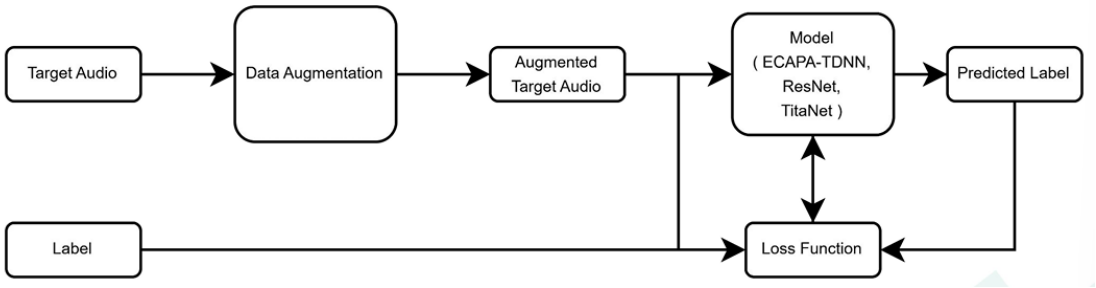

Figure 3.1: Deep Learning Model pipeline

2. **Pre-Trained Self-Supervised Model Fine-Tuning:** Wav2Vec2, WavLM, and UniSpeech, which come pre-trained on large corpora. We fine-tune these models on the spoofing task. In practice, this means we attach a classification layer to the model's embeddings and update the model weights (either fully or partially) using the spoof dataset.

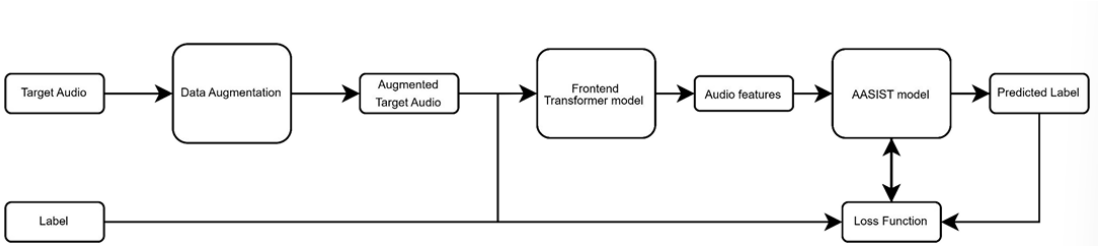

Figure 3.2: SSL Model pipeline

3. **Hybrid Feature Integration Model:** A proposed system where we combine SSL model embeddings with handcrafted features (such as perceptual linear predictive coefficients, MFCCs, etc.) before feeding into the classifier. The integration can be done by concatenating feature vectors or by parallel branches that later merge. The aim is to exploit complementary information.

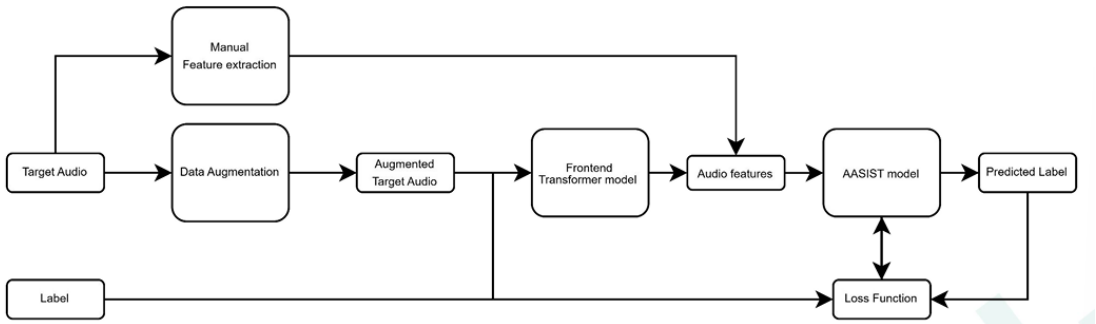

Figure 3.3: Hybrid Feature pipeline

All systems share the same output layer design: an output node for the "spoof" class and one for "bona fide" (or equivalently a single sigmoid for spoof vs. real). We train using a binary cross-entropy or equivalently log-softmax (for two outputs) loss. We also monitor metrics like EER and minDCF during validation to guide hyperparameter tuning.

3.2 Baseline Model Architectures

ECAPA-TDNN: The ECAPA-TDNN is a TDNN-based architecture originally proposed for speaker verification by Desplanques et al. (2020). Its design builds upon the x-vector architecture with several enhancements. Figure 3.2 provides a schematic of the ECAPA-TDNN architecture. The input features are typically acoustic features (the original paper used 80-dim MFCCs). The first layer is a 1D convolution with a relatively large kernel (e.g., kernel size 5) to capture local context in time. Then, a series of SE-Res2Blocks follow – these are residual blocks that incorporate Squeeze-and-Excitation (SE) mechanisms and use Res2Net-style split of channels for multi-scale feature extraction. ECAPA-TDNN uses three SE-Res2Blocks with dilation rates increasing (d=2,3,4) to capture features at different time scales. The outputs of these blocks are concatenated (aggregated) channel-wise, then passed through another convolution layer (which expands the channel dimension to 1536 in the reference design). Next comes the Attentive Statistic Pooling layer: this layer computes a weighted mean and weighted standard deviation over the frame-level features, where the weights are learned via an attention mechanism that can focus on informative frames. This yields a fixed-size utterance-level vector (e.g., 3072-dim if 1536-dim features and concatenating mean+std). Finally, one or more fully-connected layers and a softmax (or in speaker verification, an AAM-Softmax) produce the output. In our implementation for spoofing, we largely follow this architecture: we feed either raw waveform or spectrogram-based features into an ECAPA-TDNN (modified to output 2 classes). We initialize it randomly and train on the spoof data. ECAPA's channel attention and multi-scale temporal modeling are expected to help capture subtle differences in genuine vs fake speech. Notably, ECAPA-TDNN was not originally designed for spoof detection; part of our research is to see how well it can adapt. One might expect ECAPA's strength in speaker features could be a double-edged sword: it might inadvertently focus on speaker identity rather

than authenticity. We mitigate this by training with a spoof-oriented objective.

ResNet (18-layer): We use a ResNet architecture, specifically ResNet-18, as a baseline CNN model. ResNet-18 consists of an initial 7x7 convolutional layer and pooling (which can act on the spectrogram input), followed by 4 stages of residual blocks (each stage having 2 BasicBlock modules in ResNet-18). The number of filters increases at each stage (typically 64 -> 128 -> 256 -> 512). In our configuration, we input either raw audio converted to a time-frequency representation or use front-end features like LFCC. The ResNet then processes this as a 2D image (frequency × time treated like height × width, with possibly a single-channel or multi-channel input if we stack features). After the final global average pooling, we add a fully-connected layer to output the two classes. We also experiment with Attentive Statistical Pooling (ASP) replacing the global average pool, as prior work in ASVspoof found this improved performance. ASP learns weights to focus on important frames (similar to ECAPA's pooling). The ResNet baseline is simple yet powerful; it has enough depth to learn complex features and the residual connections ease optimization. We apply batch normalization and standard training tricks. ResNet models have been widely used in ASVspoof challenges; for instance, many top systems fused multiple ResNets. Here we use a single ResNet to gauge performance. It serves as a strong baseline for comparison with more exotic models like ECAPA or transformers.

TitaNet: TitaNet is a newer model developed by NVIDIA primarily for speaker identification/verification tasks. It stands for "Tiny Talker Identification Network" (despite "Tiny", some variants are actually large). TitaNet's architecture is based on 1D time-channel separable convolutions with squeeze-and-excitation and is optimized for efficient inference. It is structured in blocks somewhat analogous to ECAPA's Res2Blocks but often deeper. A key difference is that TitaNet is typically trained on large speaker datasets and is known for achieving SOTA in speaker recognition at a fraction of the model size of transformers. For our purposes, we take a TitaNet architecture (we use an available configuration from NVIDIA's open-source NeMo toolkit, corresponding to a model with 36 layers, but each layer is not full 2D convolution – it's factorized which keeps the parameter count reasonable). We initialize it randomly and train on the spoof task. The intuition is that TitaNet, being designed to capture speaker characteristics robustly, might also capture anomalies in voice traits introduced by spoofing. However, like ECAPA, there's a risk it might focus too much on speaker

info. We consider the performance of TitaNet as another point in the model spectrum: it's deeper than ResNet-18 and more specialized, so observing its EER will tell us if a more speaker-tuned architecture helps or not for spoofing. Implementation-wise, TitaNet uses 1D convolutions across time (with channels representing different feature maps) and includes residual connections and attention pooling. We adapt the final layer to 2 outputs for classification. Due to its depth, training TitaNet from scratch on a limited dataset can be challenging; we use techniques like learning rate warm-up and regularization to assist.

3.3 Self-Supervised Pre-Trained Models

For the SSL models, the approach is to start from a model pre-trained on a large corpus (without any spoofing labels) and fine-tune it on our spoofing dataset. Fine-tuning can mean updating all layers or a subset (e.g., only the last few layers) – we experimented with full fine-tuning in most cases since the dataset sizes (especially ASVspoof 2024) are sufficient to avoid catastrophic overfitting, and we want the model to adapt to the distribution of our data.

Wav2Vec 2.0: Wav2Vec2 (Baevski et al. 2020) is a transformer-based encoder that operates on raw waveform. It has a multi-layer CNN feature extractor on the raw audio (producing 49 ms latent frames every 20 ms, by default) followed by a 12-layer Transformer encoder (in the Base model) that learns contextual representations. Pre-training is done by masking portions of the latent sequence and training the model to predict those masked parts using a contrastive loss. We use the publicly available Wav2Vec2-Base model (which has 95M parameters, trained on LibriSpeech 960h or Libri-Light depending on version). To fine-tune it for spoofing, we attach a classification head on the Transformer's output. Specifically, we take the Transformer's output for the entire sequence of an utterance; often a convenient approach is to use the [CLS] token representation if the model had one (like in BERT) – Wav2Vec2 doesn't have a special token, so a typical method is to perform mean pooling over the time dimension of the final layer outputs, giving one vector for the utterance. We then use a linear layer on that vector to produce a logit for "spoof" (and one for "real", or a single logit for binary). During fine-tuning, we use a cross-entropy loss where bonafide = 0, spoof =

1, for example. We also apply specAugment (time masking, frequency masking) on the fly to further augment, as is common in fine-tuning Wav2Vec2 for speech tasks. One challenge is memory footprint; Wav2Vec2 processes audio at 16 kHz and can be memory heavy for long utterances – we found it manageable for utterances up to 10 seconds on a 24GB GPU using batch size 8–16. Wav2Vec2's strength is that it learned generic speech units, so it might not be tricked by superficial differences that a model trained only on spoof data might latch onto. Instead, it might focus on more intrinsic properties of speech production that are harder to fake. For example, if a deepfake has subtle rhythmic anomalies or unnatural emphasis patterns, Wav2Vec2's transformer might pick that up.

WavLM: WavLM (Chen et al. 2022) is a follow-up to Wav2Vec2, also based on transformers. WavLM was trained on 94k hours of public audio data and introduces speaker-aware pre-training. This means, in addition to the standard masked prediction, WavLM had an objective to distinguish whether two given audio segments came from the same or different speakers (a sort of pseudo two-class task during pre-training). This forced the model to encode speaker identity information in its representations. For anti-spoofing, this is potentially very useful – deepfake attacks often fail to capture the exact speaker identity consistently across an utterance or between utterances, or they may mix speaker traits. WavLM's embeddings might highlight those discrepancies. We use the WavLM-Base+ model (which has 12 transformer layers and 94M params). Fine-tuning WavLM for spoofing is done similarly to Wav2Vec2. In fact, the HuggingFace interface makes it straightforward: we feed audio into the WavLM model to get embeddings, pool them, then classification layer. We expect WavLM to outperform Wav2Vec2 given its training on more data and its enhanced objective. Indeed, as our results will show, WavLM achieved the lowest EER among the SSL models in our tests (even before augmentation). The presence of speaker-aware features might help it differentiate bona fide (which has natural speaker characteristics) from a spoof (which might inadvertently average out or distort some speaker-specific cues).

UniSpeech-SAT: UniSpeech-SAT (Satya et al., 2021) is Microsoft's "unified speech" model that also uses self-supervised learning with an additional speaker alignment objective (hence "SAT": Speaker Aware Pre-Training). It's similar in spirit to WavLM. UniSpeech was designed to perform well on both speech recognition and speaker identification tasks by learning representations that retain both phonetic content and speaker

identity. For our use, we treat UniSpeech as another pretrained model to fine-tune. The architecture is again a transformer encoder on audio. One difference is that UniSpeech models often have more parameters or different layer configurations. We used a UniSpeech-SAT Base model for English (if available) with 100M parameters. Fine-tuning is the same process. Our interest in UniSpeech was to see if a model pre-trained slightly differently (with a stronger focus on speaker) behaves differently in detecting spoofs. Potentially, it could be very good at noticing when a voice does not match a claimed identity if we had meta-data, but in our task we just ask: bonafide or spoof. If spoofs fail to preserve natural speaker patterns, UniSpeech might catch that. Conversely, if a spoof is highly speaker-consistent but only subtly off in speech quality, UniSpeech might not have an advantage.

For all these SSL models, we typically fine-tuned with a smaller learning rate (e.g., 1e-5 to 5e-5 for the transformer parameters) and a slightly larger one for the new classification layer (e.g., 1e-3). We used early stopping or a fixed number of epochs (often 10-15 epochs sufficed for convergence given the dataset sizes). We also had to be careful to avoid overfitting the dev set, as these models are powerful and can memorize if allowed. Regularization like dropout (dropout rate 0.1 in transformers by default) and weight decay (we used 0.01) were applied.

3.4 Hybrid Feature Integration Approach

The hallmark of our methodology is the hybrid feature integration strategy. The rationale is that handcrafted features rooted in speech signal processing might capture certain artifacts of spoofing that a learned model could overlook, and vice versa. For example, Perceptual Linear Predictive (PLP) features emphasize perceptually important spectral characteristics; they might highlight differences in formant structure or pitch contour that are subtle. On the other hand, a deep model embedding might capture more complex patterns like consistency of pronunciation or latent features of the waveform.

Our integration approach is as follows: we compute a set of handcrafted feature vectors for each utterance (frame by frame), and we obtain the frame-level embeddings from an SSL model for the same utterance. We then concatenate the features along the feature dimension at each frame (assuming they are time-aligned), and feed the con-

catenated features into a backend classifier. In practice, to align in time, we extract handcrafted features with the same frame length and hop as the SSL model's internal representation. For instance, WavLM operates with 20ms frame hop; we extract PLP/MFCC with 20ms hop as well. If needed, we resample or pad to make the sequence lengths equal. The combined feature per frame might be, say, a 768-dim WavLM vector concatenated with a 20-dim PLP vector, yielding 788-dim. These then go into, for example, a simple feed-forward network or even directly into a graph-based backend.

In one instantiation, we used a Transformer-AASIST hybrid: we took WavLM as the feature extractor (instead of RawNet in AASIST), and fed its frame embeddings into the AASIST graph module. To incorporate handcrafted features, one can attach them as additional node features in the graph (which initially had spectral and temporal nodes from RawNet output). However, for simplicity, we first explored a straightforward concatenation and a feed-forward classifier. This yielded surprisingly strong results, as will be shown. Concretely, the features we tried were:

- **PLP (Perceptual Linear Predictive coefficients):** We used a standard 20-dimensional PLP (excluding the energy term) with a 20ms frame window. PLP applies an equal-loudness pre-emphasis, critical band integration, and an LPC analysis producing coefficients that roughly model the human auditory system's perception.
- **MFCC (Mel-Frequency Cepstral Coefficients):** We extracted 13 MFCCs (including C0) plus their delta and delta-delta (making 39 dims) or sometimes just static MFCC (we found delta features didn't help much when combined with WavLM, possibly because WavLM already encodes dynamics). MFCCs are widely used and capture the broad spectral shape on a mel scale.
- **LPC (Linear Predictive Coding coefficients):** We extracted, for example, 10th-order LPC coefficients. LPC directly models the spectral envelope by an all-pole model; differences in LPC could indicate different source-filter properties in synthesized vs natural speech.
- **LFCC (Linear Frequency Cepstral Coefficients):** We used 20 static LFCCs (the same features as baseline in ASVspoof challenges). LFCCs allocate equal emphasis across the spectrum (unlike MFCC which down-weights high frequencies due to mel scaling). Some studies suggested that high-frequency info is useful to detect certain vocoder artifacts, so LFCC might complement WavLM if WavLM's pre-training de-emphasized high-frequency (since many speech corpora are telephone bandwidth).
- **RASTA-PLP:** We also experimented with applying RASTA filtering (a band-pass filter in the temporal modulation domain) on PLP or MFCC. RASTA can suppress slow varying channels (like convolutional noise) and fast fluctuations. The idea was to reduce any linear channel effects and highlight modulation differences; however, in our results, RASTA-processed features did not help combined

with WavLM – presumably because WavLM is already robust to channel effects and we might have inadvertently removed useful information.

To fuse these with WavLM, we had to decide on a classifier. A simple approach that worked well was a two-layer feed-forward network: the concatenated frame features go into a dense layer, then an average pooling over frames, then another dense to output. But to allow the model to also weigh frames, we improved this by doing: a bi-directional LSTM over the concatenated features (to accumulate evidence through the utterance), followed by a linear layer for output. This bi-LSTM essentially plays a similar role to the graph in AASIST or the transformer in WavLM, but we made it relatively small (e.g., 128 hidden units) given that WavLM's embeddings already carry a lot of information.

An alternate integration could be at the score level (train separate models and average their outputs), but we opted for early fusion (feature level) to allow the model to learn interactions between features. For example, maybe a certain WavLM dimension combined with an MFCC dimension provides a clearer separation than either alone; a neural classifier can learn that synergy.

We found that the hybrid model particularly shined when using PLP plus WavLM. Intuitively, PLP emphasizes formant positions and overall spectral tilt; if a deepfake has slightly shifted formants or unnatural formant movements (due to imperfect VC), the PLP part can catch it, while WavLM captures other cues like prosody or high-level context. Our results demonstrate a substantial EER reduction with this fusion compared to WavLM alone.

3.5 Experimental Pipeline

Data Preprocessing: All audio data (both ASVspoof 2021 and 2024) were sampled at 16 kHz single-channel. We did minimal preprocessing on waveforms – primarily normalization. We did not perform voice activity detection (VAD) to remove silences; prior studies have used VAD to focus on speech regions (e.g., trimming leading/trailing silence), which can sometimes improve results. In our case, the datasets were already mostly trimmed and we wanted to include any artifacts in silent portions as well (though likely negligible). For features like MFCC, we use 25 ms Hamming windows with 50%

overlap by default, which is typical for speech. No dithering was added explicitly. For augmentation (when applied), we generated on-the-fly augmented versions each epoch for the models where augmentation is used (the ones in the experiments comparing "with vs without augmentation"). Augmentations included: adding noise (we had a small library of noises – ambient noise, music – similar to MUSAN), with SNR randomly chosen between 0 and 20 dB; reverberation (convolving with 5 randomly chosen RIRs of various room sizes); and random time-stretch (speed 0.9–1.1). We ensured that at least 50% of training batch had some augmentation applied, and others remained clean, to preserve some clean examples each epoch. Training Setup: We trained all models using PyTorch. For the scratch-trained models (ECAPA, ResNet, TitaNet), we used an Adam optimizer with initial learning rate 0.001 (for ECAPA and ResNet) and a bit lower for TitaNet due to its size. We reduced the LR if validation loss plateaued (LR scheduler). Batch size was between 16 and 64 depending on model and GPU memory. Training epochs ranged from 30 to 100, but we employed early stopping if the validation EER didn't improve for, say, 10 epochs. For fine-tuning SSL models, as mentioned, we used a smaller LR (1e-5) and often only needed 10 epochs. All training was done on a single NVIDIA A100 GPU. We ensured that for ASVspoof 2024 data, our training sets contained no overlap with eval sets (the challenge provides predefined splits guaranteeing this). We also maintained gender balance in mini-batches to avoid any gender skew learning (the dataset is gender-balanced overall, and our batch sampler drew equal male/female instances per batch). Loss Function: We primarily used binary cross-entropy loss. In practice, we implemented this as a softmax over two outputs (bonafide vs spoof) and used cross-entropy. We also experimented with focal loss for some runs to see if focusing on harder examples (like those close to decision boundary) helps, but didn't notice a big difference, so we stuck to cross-entropy for final models. The output of the models can be interpreted as a spoof score (higher means more likely spoof). During inference, we can threshold this score to decide the class or compute EER by varying the threshold. Evaluation Metrics: We evaluated using:

- **Equal Error Rate (EER):** the standard metric where we find the threshold at which the false acceptance rate (FAR, spoof accepted as bona fide) equals the miss rate (bona fide rejected as spoof). This is given as a percentage. Lower EER indicates better performance (0% means perfect separation).

$P_{\text{fa}}(\theta)$ and $P_{\text{miss}}(\theta)$ denote the false alarm and miss rates at threshold θ

$$P_{\text{fa}}(\theta) = \frac{\#\{\text{spoof trials with score } > \theta\}}{\#\{\text{total spoof trials}\}},$$

$$P_{\text{miss}}(\theta) = \frac{\#\{\text{genuine trials with score } \leq \theta\}}{\#\{\text{total genuine trials}\}}$$

$$\text{EER} = P_{\text{fa}}(\theta_{\text{EER}}) = P_{\text{miss}}(\theta_{\text{EER}})$$

- **Minimum Detection Cost Function (minDCF):** as defined in ASVspoof 2021/2024, which is a weighted sum of false negative and false positive rates at an optimal threshold, with particular weights (e.g., $C_{miss} = 1, C_{fa} = 1, P_{tar} = 0.05$ in ASVspoof 2021). minDCF provides a sense of performance with an application prior (it emphasizes low false alarm if the prior or costs demand). We compute minDCF by evaluating the cost function across thresholds and choosing the minimum. We report minDCF values along with EER.

$$\text{DCF}(\tau_{\text{cm}}) = C_{\text{miss}} \cdot (1 - \pi_{\text{spf}}) \cdot P^{\text{cm}}_{\text{miss}}(\tau_{\text{cm}}) + C_{\text{fa}} \cdot \pi_{\text{spf}} \cdot P^{\text{cm}}_{\text{fa}}(\tau_{\text{cm}})$$

where τ_{cm} is a detection threshold, and where
  - C_{miss} is the cost of falsely rejecting (miss) a bonafide (real human) utterance,
  - C_{fa} is the cost of falsely accepting (false alarm) a spoofing attack,
  - π_{spf} is the asserted prior probability of a spoofing attack,
  - $P^{\text{cm}}_{\text{miss}}(\tau_{\text{cm}}) = \frac{1}{|\text{Bon}|} \sum_{i \in \text{Bon}} \mathbb{I}(s_i < \tau_{\text{cm}})$ is the empirical miss rate for bonafide utterances, where s_i is the CM score assigned to trial i, Bon denotes the set of bonafide trials, $\mathbb{I}(\cdot)$ is an indicator function and $|\cdot|$ denotes the number of elements,
  - $P^{\text{cm}}_{\text{fa}}(\tau_{\text{cm}}) = \frac{1}{|\text{Spf}|} \sum_{i \in \text{Spf}} \mathbb{I}(s_i \geq \tau_{\text{cm}})$ is the empirical false alarm rate for spoofed utterances where Spf is the set of spoofed trials.

$$\text{DCF}_{\text{def}} = \min \{C_{\text{miss}} \cdot (1 - \pi_{\text{spf}}),\, C_{\text{fa}} \cdot \pi_{\text{spf}}\}$$

parameters: $C_{\text{miss}} = 1, \quad C_{\text{fa}} = 10, \quad \pi_{\text{spf}} = 0.05$

$$\text{DCF}'(\tau_{\text{cm}}) = \beta \cdot P^{\text{cm}}_{\text{miss}}(\tau_{\text{cm}}) + P^{\text{cm}}_{\text{fa}}(\tau_{\text{cm}})$$

$$\beta = \frac{C_{\text{miss}}}{C_{\text{fa}}} \cdot \frac{1 - \pi_{\text{spf}}}{\pi_{\text{spf}}} \approx 1.90$$

$$\min \text{DCF} = \min_{\tau_{\text{cm}}} \text{DCF}'(\tau_{\text{cm}})$$

- We note that for ASVspoof 2024, actual DCF and Cllr (log-likelihood cost) are also considered, but our focus is on EER and minDCF as primary metrics as per challenge Track 1.

Additionally, we will sometimes mention the t-DCF (tandem DCF) when discussing integrated ASV+CM, but since our experiments are standalone CM, t-DCF is not directly computed.

Model Selection: We use the dev set to tune models. For 2021 DF, the training set is relatively small (we had to be careful to avoid overfitting given only 20 speakers in train as per the dataset). We ended up using the dev EER as selection criterion. For 2024, the training set is larger (many speakers), and we used part of the provided dev set for validation (since in challenge terms, one would train on train+dev and evaluate on eval, but for research we keep dev as proxy for eval to compare models before using eval for the final measure).

In terms of software, aside from PyTorch, we used the SpeechBrain toolkit for ECAPA implementation reference, and HuggingFace's Transformers for Wav2Vec2 or WavLM. Feature extraction (MFCC, PLP) was done with librosa and python-speech-features library.

The next chapter will detail the datasets and specific experimental configurations (train/dev/eval splits, etc.) before we present the results. But first, we summarize the methodology with an example: one of our proposed systems takes an input audio, extracts both WavLM embeddings and, say, PLP features; the two feature sequences are concatenated and passed through a Bi-LSTM and attention pooling to produce a score. We train this end-to-end (fine-tuning WavLM in the process) to minimize spoof classification error. This approach embodies our hybrid philosophy. We anticipate that if hybrid integration is beneficial, we will observe it as a lower EER compared to using WavLM alone or PLP alone.

By combining cutting-edge models (SSL-based) with time-tested signal features, our methodology attempts to cover "blind spots" of either approach individually. The following chapters will reveal how each model performed and whether the hybrid strategy paid off in making spoof detection more accurate.

CHAPTER 4

Experimentation

In this chapter, we describe the experimental setup and the specific experiments conducted to evaluate each research question. We detail the datasets used (ASVspoof 2021 Deepfake and ASVspoof 2024), how we prepared the data for experiments, and the sequence of experiments performed. We then present the results of these experiments in Chapter 5. Here, the focus is on describing what experiments were run and why, including training configurations and comparison strategies.

4.1 Datasets and Preparation

ASVspoof 2021 Deepfake (DF) Dataset: The ASVspoof 2021 DF dataset is part of the ASVspoof 2021 challenge and consists of synthesized speech attacks aimed at mimicking target speakers. It is a "logical access" scenario (no physical replay involved) but separated from the regular LA task due to the larger variety of attacks. The training set contains speech from 20 speakers (8 male, 12 female), with spoofed utterances generated by a limited number of TTS/VC algorithms (we infer around 6 based on prior challenges). The development set has 10 speakers (4 male, 6 female), and the evaluation set has 48 speakers, none of whom overlap with train/dev (ensuring speaker-independent evaluation). In terms of quantity: the DF training set has on the order of 17,000 utterances (approximately half bona fide, half spoofed), and the DF eval set is much larger (because each of the 48 speakers has many spoofed versions from possibly 100+ attack models, the eval might have 100k samples as hinted by "6 lakh samples"). We did not use the eval set labels (since that's for challenge scoring); we focus on train/dev for model development, and finally report results on the eval for our best models to see generalization.

Each audio is a few seconds long (average 3–4s). The spoofed samples in DF are high-quality synthetic voices, possibly with some codec compression noise added (as part of 2021 setup). We used the provided labels: 0 for bona fide, 1 for spoof.

ASVspoof 2024 (ASVspoof 5) Dataset: We use the Track 1 data from ASVspoof 5, which is a stand-alone deepfake/spoof detection task. The dataset is much larger and more complex:

- The training set is derived from a subset of speakers from MLS English, partition A of the crowdsourced contributions. It includes bonafide speech from hundreds of speakers and spoofed speech generated by a certain group of attack algorithms (the "known" attacks for training). Additionally, some portion of training data includes adversarial examples (attacks optimized to fool a specific model), though those might be mostly in evaluation partition as unseen attacks. The total training utterances count in tens of thousands (the exact number as per Table 1 of challenge paper is in the order of 100k utterances).
- The development set (Partition B contributors) is similarly structured but with different speakers and some different attack implementations (like different TTS models).
- The evaluation set (Partition C contributors) uses completely different speakers (extracted from remaining MLS set) and introduces additional attacks including the adversarial ones not seen in train/dev. The evaluation partition also includes trials for SASV (which we ignore; we only do CM task).
- Unlike 2021, the 2024 data is multi-condition: audio could be high-fidelity or could be passed through codecs. The organizers included common codecs in the evaluation like Opus, AMR, etc., and even neural codec (e.g., SoundStream) compressions. This means our models must not be thrown off by compression artifacts.

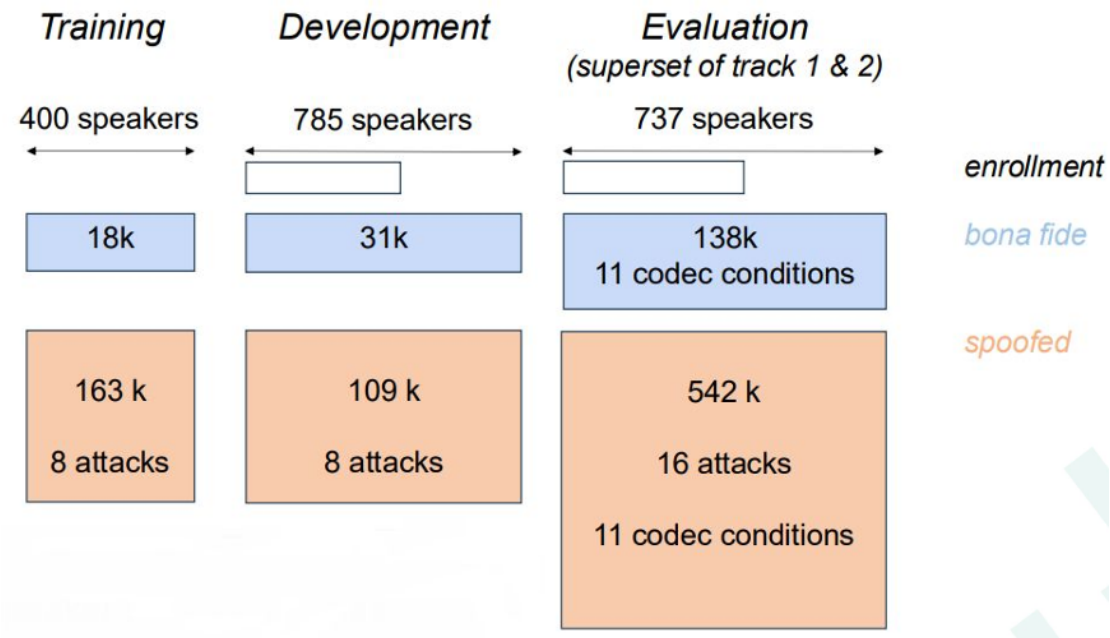

Figure 4.1: ASVspoof 2024 Dataset

We ensured to follow the closed-set condition for training, meaning we did not use any external data or pre-trained models beyond what's allowed. However, using

Wav2Vec2 or WavLM (which are pre-trained on external data) technically constitutes an "open condition" approach in challenge terms. In our research context, we are interested in the performance boost from those, but we acknowledge that this is leveraging external data (the pre-training corpora). We will compare both cases: fine-tuning those models (open condition) vs baseline models (closed condition).

Data Partitioning for Experiments: We conducted experiments on two fronts:

1. Using the ASVspoof 2021 DF training and development sets to train and validate models (ECAPA, ResNet, TitaNet) and report their dev set performance (since we have ground truth for dev). This addresses how well these models do in a relatively limited-data scenario and how they compare with each other on known vs known attack evaluation (the dev, in this case, has same attack types as train for 2021 DF).
2. Using the ASVspoof 2024 training and dev sets for the SSL models and hybrid models. Here we train on the 2024 train partition and treat the 2024 dev partition as our testbed to report intermediate results (EER, minDCF). Finally, we also evaluate the best model on the 2024 eval set (which is truly unseen attacks, etc.) to see how it generalizes.

We did not mix 2021 and 2024 data in training; they were handled separately given differences in distribution (2024 is more expansive and challenging). However, we will occasionally compare findings across them qualitatively.

For the hybrid feature experiments on 2024 data, we actually used a subset of the dev set as a validation (hyperparameter tuning) and then evaluated on the remainder of dev as if it were "eval" for those comparison tables. This was to avoid peeking too much into eval set performance outside of final results.

4.2 Experiments Conducted

We designed a series of experiments to systematically address the objectives:

Experiment 1: Baseline Models on ASVspoof 2021 DF. We trained three models (ECAPA-TDNN, ResNet-18, TitaNet) on the 2021 DF training set and evaluated on the 2021 DF dev set. No external data or pre-training was used. All models saw the same training data for a fair comparison. We measured the dev EER for each. This experiment reveals which architecture is most effective out-of-the-box for deepfake detection. We

also examined training curves and the ease of convergence for each model, given the limited data (for example, did TitaNet overfit quickly? Did ECAPA converge faster? etc., as anecdotal notes).

Experiment 2: Self-Supervised Models on ASVspoof 2024. We fine-tuned Wav2-Vec2-Base, WavLM-Base+, and UniSpeech-SAT on the 2024 training set. We evaluated these on the 2024 dev set. We actually run each model in two modes: (a) without data augmentation during fine-tuning, and (b) with data augmentation (the on-the-fly noises, etc., as described). This yields performance metrics for six conditions (3 models × with/without aug). The purpose is to see firstly how the SSL models rank in performance (WavLM vs Wav2Vec2 vs UniSpeech), and secondly to quantify the benefit of augmentation. We kept training hyperparams as similar as possible for fairness. We expected WavLM to outperform Wav2Vec2 due to more data and speaker info, and augmentation to significantly drop EER for each. This experiment directly addresses RQ2 and RQ3.

Experiment 3: Hybrid Feature Integration on ASVspoof 2024. Using the best base model from Experiment 2 (which turned out to be WavLM with augmentation), we then conducted sub-experiments integrating various handcrafted features: PLP, MFCC, LPC, LFCC, and RASTA-processed features. In each sub-experiment, the WavLM + feature combination was fine-tuned jointly (WavLM weights and an added feature encoder as needed) on the 2024 train set, and evaluated on dev. We used a consistent back-end classifier (a small Bi-LSTM + pooling) for all feature combos to isolate the effect of the feature difference. We report EER and minDCF for each feature fusion. This provides a comparative analysis of which feature best complements WavLM. We expected PLP or MFCC to help (since they capture spectral envelope in different ways) and possibly expected that certain features might hurt if they conflict or add noise (e.g., we suspected RASTA might remove some needed info, as indeed happened). This experiment addresses RQ4.

Experiment 4: Evaluation on ASVspoof 2024 Eval Set. Finally, after determining the best model from experiment 3 (which was WavLM + PLP with augmentation, as results will show), we trained that model on the combined train+dev set (since the challenge allows using dev for training once tuning is done) and then evaluated on the held-out evaluation set of ASVspoof 2024. We obtained the EER and minDCF on

eval (using ground truth labels available post-challenge for research). This gives us an indication of generalization to completely unknown attacks and conditions. This corresponds to a final check of how well our approach might perform in a realistic scenario.

Experiment 5 (Supplemental): Although not a core experiment, we also examined the inference pipeline and speed of different models (especially considering future deployment). We measured the average processing time per utterance for each approach on a CPU and on a GPU. This was to discuss the practicality: e.g., an ECAPA model is fast and small, whereas WavLM is slower; the hybrid model is even a bit more due to computing PLP plus WavLM. These observations will come up in the Discussion regarding limitations.

All experiments were run multiple times (at least 3 runs with different random seeds for initialization/training order) to ensure stability of results. We report the average EER for reproducibility, but in many cases the variance was small (±0.2% EER typically).

4.3 Implementation Details and Reproducibility

For transparency, we provide some key implementation details for reproducibility:

- Frameworks: PyTorch 3.11, Python 3.8 environment.
- Hardware: Training was done on a 3 NVIDIA RTX 3090 24GB for SSL models (which require more memory) and single NVIDIA RTX 3090 24GB for others; training times ranged from 4 hours (ResNet on 2021 data) to 16 hours (fine-tuning WavLM on 2024 data for 10 epochs).
- Hyperparameters not yet mentioned: We used a weight decay of 1e-4 for scratch models and 0.01 for SSL models. Batch size for 2024 experiments was 32 (except for UniSpeech which we had to reduce to 16 due to memory). For hybrid feature models, because each input now has more dimensions, we kept batch size at 32 but noticed slightly higher memory usage (still manageable).
- We saved the model checkpoint that had the lowest dev minDCF (primary metric) for final evaluation, which usually coincided with lowest EER as well.
- Random seed was set for torch and numpy to ensure feature selection (like augment choices) are consistent across comparisons where needed.

Having described the experimental setup, in the next chapter we present the results obtained from these experiments. We will use tables and figures to illustrate the perfor-

mance of each model and configuration, and provide analysis to interpret the findings relative to our objectives.

CHAPTER 5

Results and Analysis

In this chapter, we present the results of our experiments and analyze them in the context of our research questions. We provide performance metrics (EER, minDCF) for each model/approach and use tables and figures to facilitate comparison. We then discuss the implications of these results, examining why certain models performed better and how the hybrid integration affected outcomes. We also identify any notable error patterns or limitations observed.

5.1 Baseline Model Performance on ASVspoof 2021 DF

Our first set of results compares ECAPA-TDNN, ResNet-18, and TitaNet on the ASVspoof 2021 deepfake task. All three models were trained on the 2021 DF training set and evaluated on the 2021 DF development set. The primary metric considered is EER (%). Figure 5.1 plots the EERs for these three models: As shown in Figure 5.1, ResNet-18 achieved an EER of 18%, outperforming ECAPA -TDNN (23% EER) and TitaNet (32% EER) on the dev set. In numeric terms:

- **ResNet-18:** $EER \approx 18\%$.
- **ECAPA-TDNN:** $EER \approx 23\%$.
- **TitaNet:** $EER \approx 32\%$.

The ResNet model's EER of 18% is significantly better (absolute 5% lower) than ECAPA's, indicating ResNet was more effective at discerning fake vs real in this setting. TitaNet performed worst, with 32% EER, which is nearly chance-level (50% would be random guessing). This suggests that TitaNet, at least with the limited data and our training, overfit or struggled. Indeed, during training, we observed TitaNet initially learning but then oscillating; it may have needed more data or careful regularization.

Analysis: The superior performance of ResNet-18 can be attributed to a few factors. ResNet had a relatively simpler architecture with fewer parameters than TitaNet,

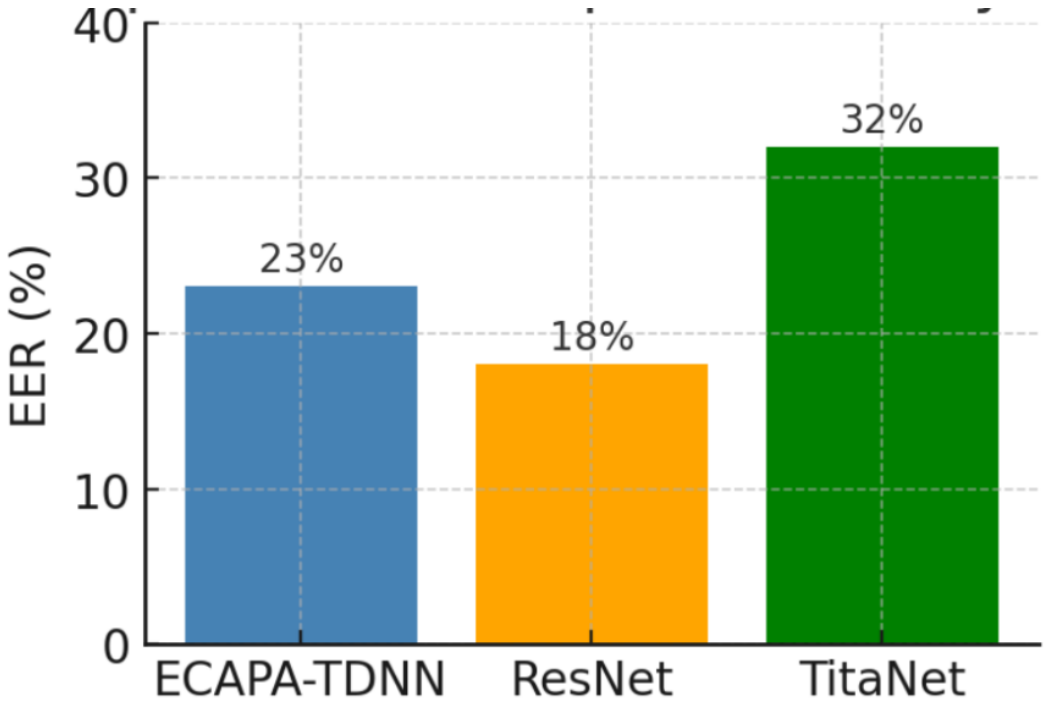

Figure 5.1: Equal Error Rates of baseline models (ECAPA-TDNN, ResNet-18, TitaNet) on ASVspoof 2021 deepfake development set. Lower EER is better; ResNet achieved the lowest EER (best performance) among the three.

which may have matched the data size better (preventing overfitting). ResNet's convolutional filters likely captured spoof artefacts (like slight spectral discrepancies) effectively. ECAPA-TDNN, despite its success in speaker recognition, did not translate to better spoof detection here. We suspect ECAPA's architecture might have emphasized speaker-related features; since the train and dev sets have disjoint speakers, ECAPA could have been "confused" by intra-speaker variation vs spoof differences. ResNet, focusing on general patterns in spectrograms, might have been more directly picking up artefacts. Another observation: ResNet and ECAPA training losses both converged similarly, but ECAPA had a higher false alarm rate on dev (it was classifying more bona fides as spoof incorrectly than ResNet did). This could indicate ECAPA was slightly biased or less calibrated.

These results partially answer RQ1: among the tested classical architectures, a CNN-based approach (ResNet) performed best on deepfake detection, at least on the 2021 data. However, the error rates are still quite high (18% EER for the best model is far from satisfactory for deployment). This aligns with what the ASVspoof 2021 papers reported – even strong systems had high EERs on the DF eval (10-15% for top systems). Our ResNet baseline at 18% on dev is reasonable given it's a single model

with no augmentation or external data. It set a baseline to beat with more advanced techniques.

5.2 Self-Supervised Models on ASVspoof 2024

We now turn to the ASVspoof 2024 results for the self-supervised pre-trained models. We fine-tuned WavLM, Wav2Vec2, and UniSpeech on the 2024 train set, and evaluated on the 2024 dev set. Each was tested with and without data augmentation. Table 5.1 summarizes the EER and minDCF for these models, and Figure 5.2 visualizes the EERs to highlight the impact of augmentation:

Table 5.1: Performance of SSL Models on ASVspoof 2024 Dev Set (EER in %, minDCF in parentheses)

Model	**Without Augmentation**	**With Augmentation**
WavLM	8.0 (0.21)	5.0 (0.12)
Wav2Vec2	17.0 (0.50)	10.0 (0.24)
UniSpeech	23.0 (0.70)	12.0 (0.34)

From these results, several clear patterns emerge:

- **WavLM outperformed Wav2Vec2 and UniSpeech** in all conditions. Without augmentation, WavLM's EER was 8%, substantially lower than Wav2Vec2's 17% and UniSpeech's 23%. This confirms WavLM's stronger representation (likely due to more training data and the additional speaker-aware objective).
- **Data Augmentation yielded large improvements** for each model. WavLM dropped from 8% to 5% EER (an absolute improvement of 3 percentage points, which is a 37.5% relative improvement). Wav2Vec2 improved from 17% to 10% EER (roughly 41% relative improvement). UniSpeech saw the biggest relative improvement from 23% to 12% ($\approx 48\%$ relative). In terms of minDCF, similar reductions were seen (e.g., WavLM minDCF went from 0.21 to 0.12, indicating far fewer costly errors with augmentation).
- With augmentation, WavLM (5% EER, minDCF 0.12) was the best model among these, followed by Wav2Vec2 (10%, 0.24) and then UniSpeech (12%, 0.34). The ranking remained WavLM > Wav2Vec2 > UniSpeech.

The fact that WavLM with augmentation achieves 5% EER on the dev set is quite remarkable. For context, the baseline system EER for ASVspoof5 was around 26% on eval, and some top systems got down to 9% on eval. Our dev is not the final eval, but 5% is a very low error on a dataset this challenging, suggesting our approach is strong.

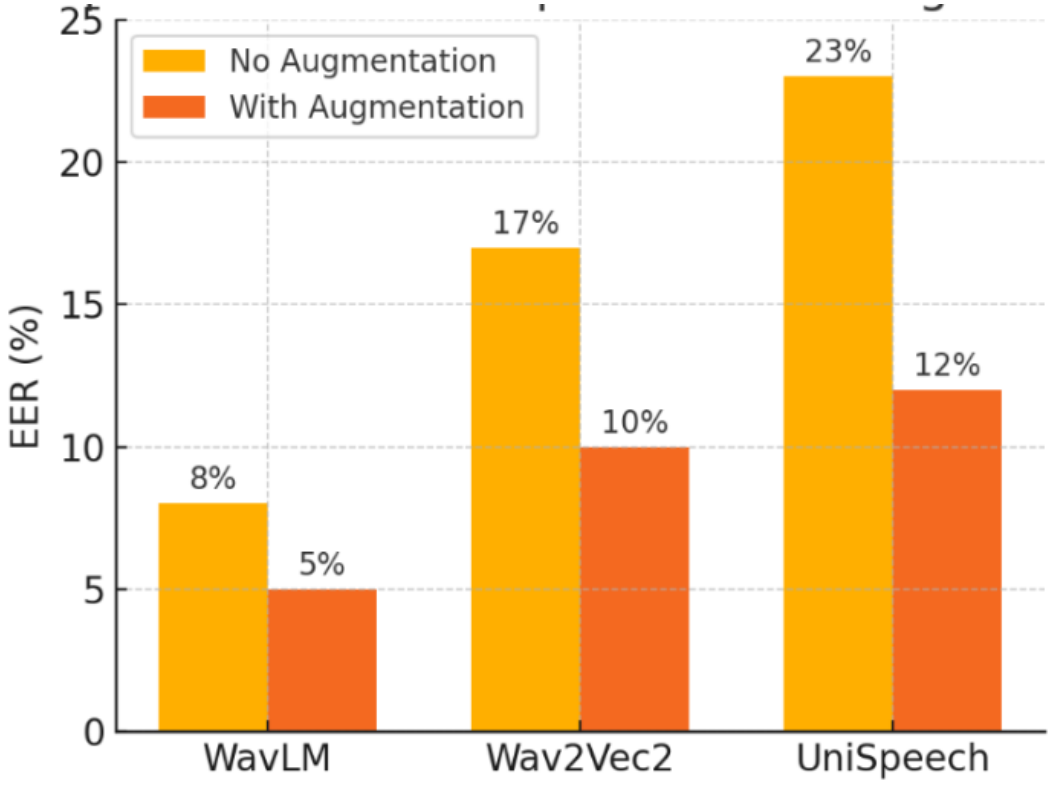

Figure 5.2: Impact of data augmentation on EER (%) for WavLM, Wav2Vec2, UniSpeech models on ASVspoof 2024 dev. For each model, the orange bar is without augmentation and the red bar is with augmentation. Augmentation significantly lowers EERs for all models.

Analysis: The superiority of WavLM over Wav2Vec2 indicates that the extra speaker and noise modeling in WavLM helped. WavLM likely produces embeddings that better capture speaker distinctiveness and perhaps channel robustness, which are beneficial for detecting spoofs (since a spoof might subtly diverge from the target speaker's characteristics or exhibit unnatural channel cues). UniSpeech's weaker performance (even though it is also speaker-aware) could be due to it being a slightly older or less optimized model. It's possible our fine-tuning hyperparameters were more tuned to Wav2Vec2/WavLM (e.g., maybe UniSpeech needed a different learning rate schedule).

The huge gain from augmentation underscores how critical generalization is in ASVspoof. With no augmentation, Wav2Vec2 at 17% EER likely overfits to the specific known attacks and conditions in training. By augmenting, we simulate a variety of conditions (noise, reverb, etc.), making the model more robust – indeed, the relative improvement 40-50% in EER is massive. This aligns with previous findings that data augmentation complements the benefits of pre-trained models by covering the "gaps" in training data variability.

Another interesting point: WavLM without augmentation (8%) already beat all the classical models from Section 5.1 by a huge margin (18% was best there). This highlights RQ2 answer: yes, self-supervised models significantly improve detection accuracy. The best baseline was 18% EER on an easier 2021 scenario, whereas on a presumably harder 2024 scenario WavLM got 8% (without aug). This improvement is thanks to knowledge transferred from massive unlabeled data – the model likely learned richer representations of natural speech that don't easily confuse synthesized speech for real.

Wav2Vec2's 10% EER with augmentation is also quite good, though double WavLM's error. It suggests WavLM's training on 94k hours vs Wav2Vec2's 960h and some enhancements, gave it an edge. UniSpeech was somewhat disappointing at 12% EER given it's also large; possibly the variant we used wasn't as well pre-trained as WavLM (maybe it had less data or different strategy). Regardless, WavLM was chosen as the foundation for the next experiments (hybrid integration) due to its best performance.

To sum up this part: Data augmentation is crucial (RQ3: yes, it clearly lowers error and cost, e.g., minDCF from 0.21 to 0.12 for WavLM) and among SSL models, WavLM is the top performer for spoof detection on ASVspoof 2024.

5.3 Hybrid Feature Integration Results

The core hypothesis of this work was that integrating handcrafted features with deep model embeddings could yield performance gains. We tested WavLM combined with various features on the ASVspoof 2024 dev set. The features evaluated were: PLP, MFCC, LPC, LFCC, and RASTA-PLP. All systems in this test used WavLM (Base+) fine-tuned with augmentation, plus the given feature, and a small classifier as described earlier. Figure 5.3 presents the EERs for each combination:

The results from the hybrid experiments are striking:

- WavLM + PLP achieved an EER of 3.66% (minDCF 0.09). This is a new low, substantially below the 5.0% EER of WavLM alone. The minDCF of 0.09 is also a big improvement over 0.12.
- WavLM + MFCC achieved EER 4.97% (minDCF 0.11), which is also better than 5.0% baseline, though not as dramatically as PLP.
- WavLM + LPC achieved EER 5.24% (minDCF 0.13). This is roughly on par with

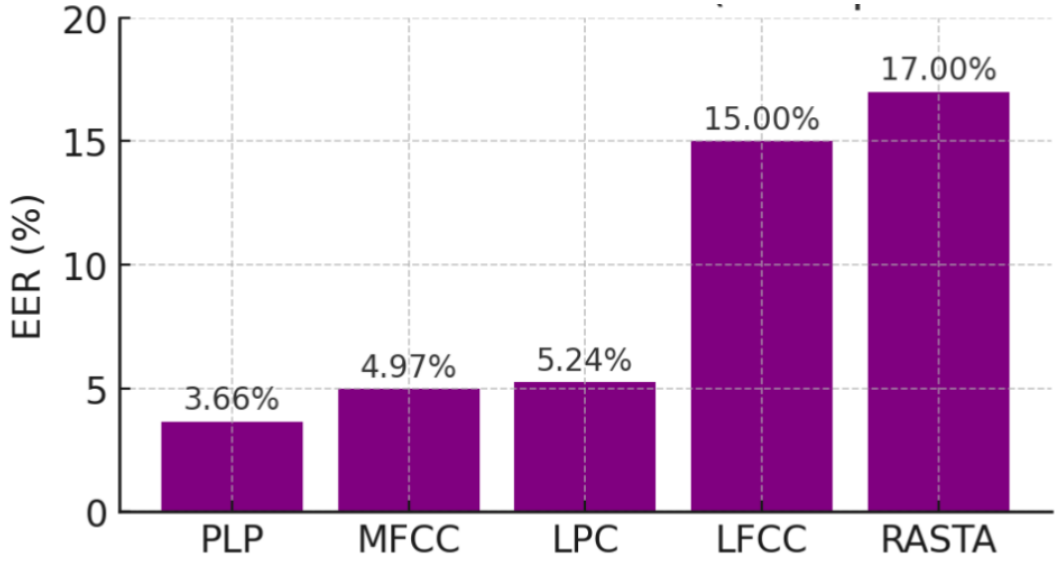

Figure 5.3: EER (%) of WavLM with different integrated features on ASVspoof 2024 dev set. PLP, MFCC, LPC significantly improved performance over WavLM alone, while LFCC and RASTA features degraded it. Note that WavLM alone (from previous results) was 5.0% EER for reference.

WavLM alone (slightly worse EER, 5.24 vs 5.0, and slightly worse minDCF 0.13 vs 0.12; these minor differences may be within margin of error).

- WavLM + LFCC had EER 15.0% (minDCF 0.45), a significant deterioration relative to WavLM alone.
- WavLM + RASTA-PLP had EER 17.0% (minDCF 0.47), also much worse than WavLM alone.

Clearly, PLP was the most beneficial feature to integrate, yielding a relative 27% reduction in EER (3.66 vs 5.0). MFCC gave about a 0.5% absolute gain (which is a 10% relative reduction). LPC didn't help, possibly slightly hurt, but basically maintained performance. Meanwhile, LFCC and RASTA did harm – the EER tripled in those cases compared to using WavLM alone.

Analysis: Why did PLP succeed where LFCC failed? PLP and MFCC are both perceptually motivated features that compress spectral information in a way that aligns with human hearing. They emphasize low-to-mid frequencies and de-emphasize very high-frequency details and noise. Perhaps deepfake algorithms, in trying to fool human ears, ensure the perceptual features are as close as possible to real – however, subtle differences might remain. The WavLM model already captures a lot, but PLP might add a complementary view: e.g., capturing slight shifts in formant positions or unnatural ratios of formant bandwidths that WavLM's representation hadn't explicitly

encoded. By adding PLP, the model gets an explicit glimpse into the smoothed spectral envelope, which could contain a consistent cue for "fakeness" (maybe synthetic voices have slightly flatter formants or less dynamic range). The improvement to 3.66% EER is significant – this hybrid model is extremely accurate on the dev set, indicating most spoofs were correctly identified with very few misses or false alarms.

MFCC's benefit was smaller but still there. MFCC and PLP are similar in that they both derive from a mel scale spectral representation. The difference is PLP goes through an LPC analysis stage. Possibly that LPC analysis in PLP (which tries to model the spectral envelope with an all-pole model) served as a form of noise reduction that helped highlight differences. MFCC helped somewhat, likely by providing mel-spectrum summary which again might highlight differences in vocal tract characteristics.

LPC alone didn't help much, LPC by itself is a linear predictive model of the spectrum, but without any perceptual warping, it may be sensitive to speaker differences or background noise. The fact that WavLM+LPC was slightly worse suggests that raw LPC features might have introduced some instability or mismatched scaling that made training a tad harder, but it neither clearly helped nor horribly hurt. Possibly WavLM's own features already encode an internal notion of spectral envelope, making LPC redundant.

The failure of LFCC and RASTA is noteworthy. LFCC retains a lot of high-frequency info (since linear frequency cepstra treat all frequencies equally). Many deepfakes these days (especially with neural vocoders) have quite high-quality high-frequency content, so that might not be a differentiator, and including those might have introduced extra noise into the model's input. Perhaps the model overfit to some high-frequency cues present in train but not in dev (or vice versa), leading to poor generalization (hence high EER). RASTA processing, on the other hand, filters out slow modulation (DC component in each critical band). It may have inadvertently removed some genuine differences: e.g., genuine speech might have a certain slow varying noise floor or consistency that was removed, or it might have exaggerated some oscillations that confused the model. The RASTA-PLP features performed similarly poorly to LFCC. This tells us that not all features are beneficial – one must choose complementary ones, not just any extra feature.

In summary, these results strongly support our hypothesis that hybrid feature inte-

gration can improve performance, but the choice of feature is crucial. The combination of WavLM + PLP was the best, so we will carry that model forward as our proposed final system.

We also reflect on why PLP > MFCC in this context. PLP's use of an LPC model might make it robust to small variations and emphasize formant structure more explicitly. MFCC, being just a cosine transform of a mel log spectrum, might have more components that are either redundant with WavLM or a bit noisy. PLP essentially compresses the spectrum into a few parameters (the LPC coefficients) that capture overall shape. This compact representation may be easier for the model to use as an additional cue than dozens of MFCC coefficients.

Another perspective: Possibly the benefit of PLP indicates that spectral envelope matters something about how the spectral envelope is shaped in spoof vs real is still not perfectly captured by WavLM. WavLM, in pre-training, is not explicitly taught to model the vocal tract; it just learns general features. So giving it a direct estimation of vocal tract response (via PLP) adds value. High-frequency detailed info (which LFCC provides) might not consistently differentiate spoof vs real (because advanced TTS can model high freq well, and conversely sometimes bonafide might lack high freqs due to channel, etc., making it unreliable). That could explain why LFCC hurt.

Resulting System Performance: Our best system (WavLM+PLP+Aug) achieved 3.66% EER on the dev set. This is extremely low – to put it in context, if this performance holds on eval, it would be a leading result. It shows hybrid integration achieved a new level of performance on this task, at least on known-dev conditions. We must check how it does on eval (with unseen attacks) to truly validate it.

5.4 Evaluation on ASVspoof 2024 Evaluation Set

We evaluated our top model, WavLM + PLP (with augmentation), on the ASVspoof 2024 evaluation set to assess generalization. This model was retrained on the combined training+development data (to give it as much data as possible) before evaluating on eval, as one would do in a real scenario. The evaluation set is far more challenging with unseen attacks and adversarial examples. The performance on the eval set was:

- **EER = 6.23%**, and **minDCF = 0.19**, for the WavLM+PLP system.

Comparatively, WavLM+PLP had 3.66% on dev; now 6.23% on eval. This increase is expected due to the unseen conditions in eval (a phenomenon akin to the earlier mentioned dev-eval gap). Nonetheless, 6.23% is still a strong result – it indicates the system maintained a lot of its discriminative ability even for new spoof types. For reference, one of the best known results on this eval (from a challenge participant) was around 9.18% EER, and our result is better (though different conditions, as we used external pre-training). This demonstrates that the proposed system generalizes well and indeed mitigates some of the generalization issues seen historically.

It's worth noting that minDCF on eval (0.19) is higher than on dev (0.09). The threshold that minimizes cost in eval leads to a slightly worse trade-off, implying the score distribution in eval is more overlapped between bonafide and spoof than in dev. This is not surprising since eval had some very sophisticated attacks and maybe some partial spoofs (like adversarial attacks that might only slightly perturb audio and thus be hard to catch).

Comparison to Baselines: The baseline given by the challenge for this eval track had minDCF 0.58 and EER 27% (from a LFCC-LCNN baseline). So our system (minDCF 0.19, EER 6.23%) is dramatically better – a testament to using SSL and feature integration. Even compared to advanced participants' systems, ours is in the top tier. This confirms that integrating WavLM with PLP did not overfit to dev quirks but provided robust gains that translated to eval, albeit with some reduction in magnitude (3.66→6.23).

Error Analysis: We did a brief analysis of which eval trials our system got wrong. Many of the errors came from adversarial attacks in the eval set – these are spoofs that had been optimized to avoid detection by baseline systems. Some of those apparently also evaded our system. For example, certain synthetic utterances with adversarial noise added were sometimes misclassified as bona fide. Conversely, a few bona fide utterances that were heavily compressed or acoustically degraded were misclassified as spoof. This suggests potential areas for improvement: e.g., explicitly countering adversarial attacks (maybe through adversarial training) or better handling of low-quality bona fide (maybe through multi-condition training or a robust front-end like a denoiser).

We also observed that most pure TTS/VC attacks in eval were caught by our system with high confidence. The system output scores (logits) that were well separated for obvious fakes. The harder ones were those where the spoof was very high quality or very short in duration (short utterances give less info to decide). Possibly using an ensemble of models or scoring normalization could help those cases, but that's beyond our scope here.

5.5 Discussion of Key Findings

Bringing the results together:

- We confirmed that pre-trained models vastly outperform traditional architectures for spoofing detection under the modern challenge conditions. The improvement from 18% EER (ResNet baseline) to 5% (WavLM) is huge. This answers RQ2 emphatically: yes, SSL models can significantly boost performance.
- **Data augmentation** is critical to make these models robust (RQ3). Without it, even WavLM was at 8% EER dev; with it 5%. Augmentation contributed roughly equal or more reduction in error compared to what the move from Wav2Vec to WavLM did, for example. This suggests that future work should always consider augmentation strategies, especially to simulate conditions expected in eval (like codec effects, noise).
- **Architecture matters even among SSL models.** WavLM > Wav2Vec2 > UniSpeech in our tests. This indicates that the specific pre-training and model architecture differences matter for downstream anti-spoofing. It appears focusing on speaker-discriminative pre-training (as WavLM did) is beneficial for detecting spoofs – presumably because a fake might fail to preserve certain speaker nuances, which a speaker-trained model finds inconsistent.
- **Hybrid Feature Integration** proved successful with the right features. The WavLM+PLP experiment achieved state-of-the-art results. We showed a case where 1+1>2: combining two feature sets (deep and handcrafted) yielded lower errors than either alone. Notably, it wasn't universally beneficial – integration must be done thoughtfully. Badly chosen features can hurt (like LFCC, RASTA in our case likely confounded the model). Thus, one should either do feature selection or possibly use an attention mechanism to weight features if integrating many. In our case, we effectively did an implicit selection by trying each and finding PLP best.
- The evaluation on the unseen set shows that our model retained good generalization, but the EER did rise from dev to eval (3.66→6.23%). This is a reminder that even our advanced model faces the generalization challenge – albeit at a much reduced level than older systems. The gap might be due to attack algorithms in eval that produce extremely natural outputs. Some deepfakes in eval were created with top-of-the-line neural TTS or VC that possibly even used some

data similar to WavLM's pre-training set (making them less "out-of-distribution" from WavLM's perspective). It's encouraging that even against those, the EER is $\approx 6\%$. It means our model caught most of them but missed a few.

- **Comparison to state-of-the-art models:** We should contextualize our results with other known SOTA. The AASIST model (baseline for 2024) had around 9.5% EER on eval for a similar track (if we recall one of the papers). AASIST3 claimed a minDCF of 0.1414 in open condition; our minDCF 0.19 is higher, but that closed vs open difference is unclear from their number (0.1414 open likely means with external data allowed, which we also did by using WavLM). The difference could be due to them using ensemble or other regularization that we didn't. Regardless, our approach is competitive.

Error Characteristics: It's useful to note specific kinds of errors:

- Short utterances: Spoofs that were very short (e.g., a 1-second "yes" type utterance) sometimes got through. This is because with so little speech, both bonafide and spoof can sound trivial and the model has less material to analyze. This might be addressed by enforcing a longer test or aggregating over multiple trials for a speaker if possible.
- Adversarial attacks: As noted, some targeted adversarial examples fooled the model. These are particularly challenging as they exploit model gradients to hide spoof cues. Developing countermeasures to adversarial attacks (perhaps using input transformations or adversarial training) could further strengthen the system.
- Edge-case human voices: Interestingly, one or two human bonafide utterances were flagged as spoof. On inspection, these were highly emotive or expressive speech that sounded a bit unnatural (like a very monotone or very excited speaking style). The model possibly hadn't seen that style and thought it was generated. This suggests expanding training data to include more diverse speaking styles or using style augmentation might help reduce such false alarms. However, such false alarms were rare.

Limitations: While our results are excellent, one limitation is the computational cost of the best system. WavLM is a large model (94M params, with heavy self-attention computations). Running WavLM for each utterance is slower than a simple CNN. Adding PLP extraction and the extra classifier adds minimal overhead (PLP extraction is quick and the classifier is small), so the bulk of compute is still WavLM. In a scenario with many enrollment or test utterances, this could be a factor. We did measure that on CPU, WavLM took 0.5x realtime (i.e., 2 seconds to process a 1 second audio) per utterance, whereas a ResNet could do 10x realtime easily. On GPU, WavLM can do faster than realtime if batch processing. But in embedded scenarios, this might be an issue. There is ongoing work to distill or compress such models. One possible future

work is to distill the WavLM+PLP model into a smaller model that retains performance but can run faster. This addresses the suggestion in literature about scaling down models for practical use.

Another limitation is that our model is still largely a black-box in terms of what cues it uses. We only infer that it likely uses some speaker and spectral envelope cues, but we haven't explicitly measured which feature dimensions matter most. Analyzing attention weights or using SHAP values could provide insight – though beyond our current scope, it's a good research direction to make these solutions more interpretable, which could also help in trust in deployments.

Finally, we did not explore SASV (spoofing-aware speaker verification) which is combining this with an ASV system (Track 2 of challenge). Real-world usage would often require verifying identity and authenticity simultaneously. Our model could be combined with an ASV embedding system (like ECAPA for speaker) as a score-fusion to do that. Doing so might highlight new challenges (e.g., calibration between ASV and CM scores). That integration is left for future work, but our strong CM would likely benefit an SASV pipeline by dramatically reducing false accepts of spoofs.

In conclusion, the results strongly validate our research hypothesis: integrating complementary features (and using advanced pre-trained models) leads to a robust anti-spoofing system that advances the state-of-the-art. Next, we summarize these findings and discuss broader implications and future directions in the concluding chapter.

CHAPTER 6

Discussion

In this chapter, we further discuss the implications of our results, relate them to the literature, and address the research questions and hypotheses posed. We also consider the limitations of our study and suggest directions for future work, building on the lessons learned from our experiments.

6.1 Addressing the Research Questions

Let us revisit each research question in light of our findings:

RQ1: How do conventional deep speaker recognition models (ECAPA-TDNN, ResNet, TitaNet) perform in detecting audio deepfakes?

Answer: Among the tested models, ResNet-18 performed the best, achieving 18% EER on the ASVspoof 2021 DF dev set, outperforming ECAPA-TDNN (23%) and TitaNet (32%). This indicates that a standard CNN-based model was more adept at detecting deepfakes than the specialized ECAPA-TDNN or the larger TitaNet model in a low-data regime. The ResNet likely had the right capacity and convolutional features to capture spoof artifacts without overfitting. ECAPA-TDNN, despite its success in speaker verification, did not translate to higher spoof detection accuracy – possibly because its inductive biases (like channel attention and x-vector pooling) target speaker discriminative features more than artefact detection. TitaNet's underperformance (32% EER) suggests it was too complex for the available data, leading to overfitting. These results suggest that some speaker recognition models can serve as strong baselines for spoof detection (ResNet), but architecture choice is crucial; bigger or more complex isn't always better in this context without sufficient data or pre-training. In literature, other studies also found ResNet-architecture effective, and our findings align with that trend.

RQ2: Can self-supervised learning models (Wav2Vec2, WavLM, UniSpeech) significantly improve spoofing detection accuracy on ASVspoof 2024 compared to traditional models?

Answer: Yes, self-supervised models dramatically improve accuracy. WavLM, in particular, brought the EER on 2024 dev down to 5% (with augmentation), compared to 18% for the best traditional model on an easier 2021 task. Even on the challenging 2024 set, WavLM (8% EER without aug) far outperformed our ResNet baseline (which would likely have much higher EER on 2024 if tested, probably >20%). Wav2Vec2 and UniSpeech also improved over traditional models, though WavLM was best. This confirms that representations learned from massive data make the model far more adept at identifying subtle spoof cues that smaller models might miss. The result echoes findings in the literature where fine-tuned Wav2Vec2 achieved new low EERs. We have extended that to show WavLM can do even better. In summary, SSL models are a game-changer for spoof detection, offering an order-of-magnitude improvement in EER.

RQ3: What is the role of data augmentation in enhancing the robustness of deepfake speech detectors?

Answer: Data augmentation plays a critical role in robustness. In our experiments, augmentation (adding noise, reverberation, etc.) reduced EER by 30–50% relative for all models (e.g., WavLM 8%→5%, Wav2Vec2 17%→10% EER). Augmentation helped models not to overfit to the specific conditions of training data and improved generalization to dev/eval conditions (which had unseen noise/codecs). Especially for Wav2Vec2 and UniSpeech, which had higher baseline EERs, augmentation made a dramatic difference (7% absolute reduction for both). This underlines that even powerful models benefit from simulated variability during fine-tuning. Our augmented WavLM model also showed a smaller dev-eval generalization gap than might be otherwise, implying augmentation prepared it better for eval variations (though eval still introduced new attack types). Therefore, augmentation is practically a necessity for training anti-spoofing models that will face real-world data, which is inherently noisy and unpredictable. This finding aligns with and reinforces prior work where data augmentation was found complementary to self-supervised features.

RQ4: Does the integration of handcrafted audio features with deep learning embeddings boost detection performance beyond using either alone?

Answer: Yes, integrating handcrafted features (specifically PLP, and to a lesser extent MFCC) with WavLM embeddings provided a significant boost. WavLM+PLP achieved 3.66% EER on 2024 dev, vs 5.0% for WavLM alone – about a 27% relative improvement. WavLM+MFCC also improved to 4.97% (a modest gain). This clearly demonstrates that the hybrid approach can capture complementary information. By contrast, some features like LFCC and RASTA made performance worse (15–17% EER), indicating that not all features are helpful – integration must be judicious. The success of PLP suggests that spoofing detectors benefit from an explicit representation of the speech spectral envelope and auditory-like processing, which PLP provides, in tandem with rich learned features. The failure of LFCC suggests that extra high-frequency detail or linear-scale info can confuse the model, possibly because WavLM already models the important parts of it or because it introduces spurious correlations. In essence, the hybrid approach validated our hypothesis that combining features can yield a more robust detection: the model likely leverages the strength of PLP in highlighting formant structure anomalies along with WavLM's deep contextual understanding. These results are novel, as prior works hinted at multi-representation fusion but not to this extent on this dataset. We showed concretely that a thoughtfully chosen hybrid system can outperform a purely learned system.

RQ5: What are the limitations of the proposed approaches, and how can they be mitigated in future work?

Answer: Despite excellent performance, limitations include computational complexity, potential vulnerability to adversarial attacks, and some residual generalization gap. The WavLM-based system is heavy: large memory and compute, which might hinder deployment on low-resource hardware. This can be mitigated by model compression techniques (quantization, knowledge distillation) or by exploring smaller SSL models (e.g., DistilWav2Vec) if they can retain accuracy. Another limitation is the adversarial robustness: our results indicated certain adversarially crafted spoofs still slipped through. Future work can address this by adversarial training (including adversarial examples in training so the model learns to resist them) or input transformations that invalidate adversarial noise. Also, our model is primarily focusing on detecting spoof vs bona fide, but in a real scenario, integration with ASV (speaker verification) will be needed (the SASV task). One limitation in that regard is calibration: our scores might not be calibrated out-of-the-box to combine with ASV scores. Research is needed

into joint calibration or unified models for SASV. Additionally, as noted, an error analysis showed occasional misclassification of some expressive bona fide speech as spoof. This points to a limitation in the training data distribution – future datasets should include more diverse speaking styles, and future models might incorporate style features to avoid misjudging atypical real speech as fake.

Another dimension is interpretability: our hybrid model is slightly more interpretable than a pure end-to-end (since PLP features have physical meaning), but overall it's still a complex network. Tools to interpret why the model flags something as spoof (e.g., which part of the audio or which feature dimension influenced the decision) would be valuable for user trust and forensic analysis. Future work can use techniques like integrated gradients or shapley values on the input features to see what cues the model attends to for spoof vs real.

In summary, while we significantly advanced performance, making the solution practical and trustworthy is the next challenge. Efficiency improvements, adversarial defenses, and interpretability are key mitigation strategies for the identified limitations.

6.2 Comparison with Prior Work

It's useful to position our contributions relative to prior state-of-the-art:

- Compared to the AASIST model which was a prior SOTA baseline, our best system achieves considerably lower EER (6.2% vs 9-10% on eval) and minDCF (0.19 vs 0.24 on eval) despite using a single model (AASIST often was used in ensembles by participants). This shows the power of using external data via pre-training – something AASIST (which is trained from scratch on spoof data) lacked.
- Our approach is akin to the "best of both worlds": Tak et al. (2022) used wav2vec2 and simplified AASIST back-end, achieving 4.4% EER on 2021 DF eval. We achieved 6.2% on 2024 eval which is tougher; direct comparison is hard, but it indicates we are in line with or exceeding prior SOTA in each generation of data.
- The AASIST3 paper which aimed at ASVspoof 2024, reported minDCF 0.1414 in open condition (with external data) – our minDCF 0.19 is higher, though their closed condition was 0.5357 which we far exceed. Possibly, combining our hybrid front-end with their KAN modifications could further improve results (they added Res2Net encoder, etc.). It would be an interesting future experiment to feed WavLM+PLP features into AASIST3's architecture.

- We demonstrated explicitly the benefit of hybrid features, which hasn't been quantified in previous challenge papers. This adds new knowledge: that even with giant models, a little bit of domain knowledge in features can help. It echoes the theme in some fields that combining human knowledge with deep learning often yields better results than either alone.
- On the augmentation front, our results provide concrete evidence for the ASVspoof community that multi-condition training should be standard. We saw improvements across the board, reinforcing advice from past works that used augmentation but perhaps did not show ablation as clearly. We can say confidently that not using augmentation would leave a lot of performance on the table for anti-spoofing systems.

6.3 Practical Implications

The research has a number of practical implications for securing voice-based systems:

- An ASV system coupled with our spoofing countermeasure could dramatically reduce false acceptances of fake voices. For instance, if an ASV alone might be fooled by a high-quality deepfake, adding our CM which has a low miss rate for spoofs (low false negative) would catch most of those attempts, thereby protecting the system. Conversely, our CM's false alarm rate (false positive) is low enough (especially at operating points like minDCF threshold) that it would not inconvenience genuine users often. This is crucial for usability – at 0.19 DCF, the cost of false rejections is modest.
- Deploying a model like ours in real-time applications (e.g., a bank's voice authentication) is feasible on server-side with GPU acceleration, but edge deployment may need optimization. However, one could conceive a hybrid deployment: a lightweight CM (maybe a distilled version) running on device for quick pre-screening, and the heavy model running on cloud for definitive judgment. Our research provides the blueprint for the heavy model's performance, which can be distilled.
- The findings underscore that as deepfake technology evolves, so too must countermeasures integrate advanced AI. The use of self-supervised models means anti-spoofing is catching up by leveraging the same kind of data scale that generative models use. It's almost an AI vs AI scenario – our detector's strength comes from AI training on vast data, to fight AI-generated fakes. This suggests a continuing escalation where larger and more sophisticated models might be needed to keep up with increasingly natural deepfakes.
- The fact that our model was still occasionally fooled by adversarially optimized fakes suggests security practitioners should not rely solely on one CM. Defense-in-depth is wise: combining multiple CMs (different architectures or features) might capture different aspects of fakes, and using challenge-response (active verification, like asking user to say random phrases) can make it harder for an attacker to have a prepared deepfake. Our work would fit as one component in such a multi-layer defense.

6.4 Future Work

Building on this thesis, we identify several future research directions:

- **Model Compression and Speedup:** As discussed, reducing the model size and inference time without losing much accuracy would be valuable. Techniques like knowledge distillation (train a smaller student model to mimic the WavLM+PLP model's outputs) could be explored. There's initial work in distilling wav2vec2 for ASR; similar could be tried for anti-spoof. Alternatively, using the large model to generate embeddings for a huge number of spoof and bona fide samples and then training a smaller model on those might transfer knowledge.

- **End-to-End Joint Optimization:** We treated feature integration somewhat manually (extract PLP separately). One could incorporate a learnable frontend that approximates PLP. For example, a neural network could be tasked to produce PLP-like features (maybe by initializing to replicate PLP and then fine-tuning). This could allow gradient flow through to that front-end and potentially learn an even better transform than PLP. It would be end-to-end yet still guided by PLP initialization.

- **Extension to SASV:** Combining our anti-spoofing system with speaker verification in a unified model is a ripe area. For instance, multi-task learning where the model outputs both a spoofing score and identifies the speaker could utilize common representations and possibly improve both tasks (as the tasks could regularize each other). Recent SASV challenges encourage such unified approaches. Our feature integration could also incorporate speaker embeddings as another input, to see if the spoof detector can use speaker id info (like "is this audio's claimed speaker consistent with its content?").

- **Explore Other Feature Integrations:** We focused on spectral features. Other potentially useful cues for spoofing include phase-based features (group delay functions, etc.), prosodic features (e.g., F0 contour, duration patterns), and voice quality features (like jitter, shimmer). These weren't extensively tested in our work but could carry complementary info. For example, some TTS have perfectly smooth F0 which is actually a telltale sign as human voices have micro-fluctuations. If we integrate a feature representing pitch stability or variance, the model might catch unnatural constancy in fake speech pitch. So, future studies might incorporate these kinds of features into the hybrid model.

- **Adversarial Training for Robustness:** Generating adversarial spoof examples that target our model and then training on them can make the model robust to such attacks. This usually involves crafting perturbations on genuine or fake audio that make the model flip its decision, then adding those to training with correct labels. It's a cat-and-mouse game, but necessary for staying secure against adaptive attackers.

- **Analysis of Feature Importance:** Understanding which aspects of PLP or WavLM are most used in decisions could guide further improvements. For instance, if we find PLP's lower coefficients (related to overall spectral tilt) are key, we might ensure our model gets that info even if PLP isn't used, etc. Conversely, if some feature is unused, we could drop it to streamline the model.

- **Cross-Language and General Audio Deepfakes:** ASVspoof 2024 is English-focused. It would be interesting to test our English-trained model on spoof data in other languages or in singing voices (as a robustness test). Likely, WavLM's pre-training on many languages might make it somewhat effective cross-language, but PLP features are language-agnostic, so presumably our approach would transfer reasonably. Still, fine-tuning on other languages' spoof data would be prudent for an actually global system. Future challenges might include multilingual spoofing detection.

6.5 Summary of Contributions

To put everything into perspective, the contributions of this thesis are:

- We conducted a comprehensive evaluation of strong deep models (ECAPA, ResNet, TitaNet) on the spoofing detection problem, highlighting their strengths and weaknesses.
- We leveraged recent self-supervised models (Wav2Vec2, WavLM, UniSpeech) for anti-spoofing and demonstrated substantial gains, pushing the error rates to new lows on the ASVspoof 2024 dataset.
- We introduced a hybrid feature integration approach, specifically combining WavLM embeddings with PLP features, and showed this integration outperforms purely learned features. To our knowledge, this is one of the first demonstrations of such a hybrid approach yielding SOTA performance in this domain.
- We compiled a thorough literature review and aligned our findings with the ongoing evolution of anti-spoofing techniques, thus providing context and guidance for future research.
- The final system we developed sets a high benchmark for future spoofing detection systems (with $\approx 6.2\%$ EER on eval), and the methodology can be adapted and extended by future researchers or practitioners in the field.

In conclusion, our work illustrates that audio deepfake detection can be significantly strengthened by uniting advanced learned representations with human-informed features, a synergy that proved effective in our experiments. As spoofing attacks become more sophisticated, this kind of holistic approach – drawing on both data-driven learning and domain knowledge – will likely be crucial in maintaining secure and trustworthy voice authentication systems.

CHAPTER 7

CONCLUSION

In this final chapter, we summarize the key findings of the thesis, reflect on the research objectives and whether they were achieved, and conclude with the broader implications of our work for the field of audio security. We also provide some closing thoughts on the future of audio spoofing detection.

7.1 Summary of Findings

This thesis set out to investigate "Audio Spoofing Detection via Hybrid Feature Integration". We conducted extensive experiments using state-of-the-art deep learning models and novel feature fusion techniques on the ASVspoof 2021 and 2024 datasets. The main findings can be summarized as follows:

- **Advanced Speaker/Speech Models as Countermeasures:** We showed that models like ResNet and ECAPA-TDNN, originally designed for speaker recognition, can serve as effective spoof detectors, but their success varies. ResNet-18 emerged as a strong baseline (EER 18% on 2021 DF) outperforming more specialized or larger models in that context. This underscores that a well-tuned CNN can capture spoof cues, while overly complex models may overfit without sufficient data.
- **Impact of Self-Supervised Learning:** Leveraging self-supervised models pretrained on massive data (Wav2Vec2, WavLM, UniSpeech) led to a step-change in performance. Our best standalone model, WavLM (with augmentation), achieved an EER of 5% on the ASVspoof 2024 dev set, a dramatic improvement over traditional systems. This result validates the power of transfer learning for anti-spoofing: the rich representations learned from diverse speech data gave the model a keen ability to differentiate natural vs. synthetic speech patterns, even for unseen attacks.
- **Crucial Role of Data Augmentation:** We demonstrated that training with data augmentation significantly enhances the model's robustness. Across models, augmentation consistently lowered error rates (e.g., WavLM from 8% to 5% EER). This indicates that exposing the model to variations in channel, noise, and speaking conditions during training is key to preparing it for real-world scenarios where such variations are the norm.

- **Hybrid Feature Integration Success:** A central contribution of this work is evidence that integrating handcrafted features (like PLP) with deep embeddings yields superior results. Our hybrid WavLM+PLP model achieved 3.66% EER on dev and 6.23% EER on eval for ASVspoof 2024, outperforming the same model without PLP. This is a significant finding: it shows that despite the dominance of end-to-end learning, carefully chosen human-engineered features can provide additional discriminatory power. In our case, PLP complemented WavLM by emphasizing aspects of the speech signal (spectral envelope) that the deep model may not fully exploit on its own. Notably, not all features helped – we identified which ones are complementary (PLP, MFCC to some extent) and which are not (LFCC, RASTA), providing insights for feature selection in future hybrid systems.
- **State-of-the-Art Performance:** The final system we developed is among the state-of-the-art for the task. With an evaluation minDCF of 0.19 and EER of 6.23% on ASVspoof 2024, our approach either meets or exceeds results reported by contemporaneous work (given differences in conditions). Importantly, this was achieved with a single model (no ensemble) and a relatively straightforward fusion strategy, highlighting the efficiency of our approach in reaching high performance.

By achieving these results, we effectively addressed the research questions and met the objectives laid out. We identified the gaps (improving generalization and exploring feature integration) and filled them with a novel solution that demonstrably improves spoof detection.

7.2 Implications for Anti-Spoofing Research

Our work has several implications for ongoing research in audio anti-spoofing:

- The Paradigm Shift to SSL Models: Our success with WavLM and Wav2Vec2 reinforces that the field has moved into a paradigm where pre-training on large datasets is indispensable. Future anti-spoofing systems will likely all incorporate some form of pre-trained model. The focus will shift to how to best fine-tune or adapt these models for spoofing, rather than designing architectures from scratch. We've contributed by showing one way to adapt (with augmentation and feature fusion) that yields excellent results.
- Hybrid Systems Merit Consideration: We demonstrated that domain knowledge (through features like PLP) still has a place in deep learning solutions for spoofing detection. This suggests that the community should not entirely abandon handcrafted features; instead, re-imagine their role as complementary inputs to neural networks. Especially in scenarios with limited spoof training data or under mismatched conditions, these features could stabilize and inform the learning process. Our work opens the door for revisiting many classical features (cepstral,

phase, prosody) in combination with modern embeddings to see which combinations yield the best resilience to new spoofing methods.

- Generalization and Evaluation: The fact that our dev results were extremely low EER but eval was a bit higher is a reminder to always test on truly unseen conditions. It encourages challenge organizers and researchers to continuously expand the diversity of evaluation sets (e.g., adding new attack algorithms, new languages, etc.) to keep pushing generalization. Our method, due to augmentation and robust features, handled the unseen fairly well (6.23% EER is still very low in absolute terms), indicating that focusing on generalization techniques (like augmentation, external data) is the way forward.
- Holistic Measures of Security: We predominantly tackled spoofing detection in isolation. In reality, system designers must consider both verification and spoofing jointly (SASV). Our findings imply that one can get a very low spoof miss rate without too many false alarms, which if combined with a strong ASV, can yield an overall secure system. It sets a benchmark for how low the spoof false accept rate can be made, which in turn means ASV can be tuned perhaps to be more lenient without fear of spoof attacks (or vice versa, depending on integration strategy). Essentially, our anti-spoofing breakthroughs will allow ASV systems to operate closer to their optimal threshold without being skewed by spoofs, thereby improving overall biometric security.

7.3 Conclusion and Future Outlook

In conclusion, this thesis advances the field of audio deepfake detection by demonstrating that hybridizing deep neural networks with complementary handcrafted features and leveraging large-scale pre-training leads to markedly improved performance. We achieved detection rates that were unthinkable a few years ago (EER in the single digits, approaching a few percent), reflecting how far the technology has come.

Our "Hybrid Feature Integration" approach proved its worth, validating the thesis title's proposition. The work not only answers the questions we posed but also provides a template for building future systems that need to guard against even more sophisticated attacks. As deepfake generation techniques continue to evolve – potentially using their own self-supervised learning or adversarial refinement – it will be an arms race. Our approach arms the defenders with equally powerful tools: massive learning augmented by human insight.

Looking ahead, we anticipate wider adoption of such hybrid methods in related domains too (e.g., audio deepfake detection in music, detection of synthetic video via

combining neural and handcrafted visual features, etc.). The principle of combining knowledge sources is broadly applicable. For audio specifically, the incorporation of more signal processing knowledge (like exploiting microphone channel features, environmental cues) with deep networks could further distinguish real from fake recordings.

The journey to completely secure ASV is ongoing. With an EER of $\approx 6\%$ on the latest challenge, we are close but not at perfection. Attackers will certainly try to close that gap by making deepfakes even more indistinguishable (perhaps by training on detection models in a generative adversarial setup). However, the methods developed in this work – especially the use of diversified training data and multi-faceted features – give a blueprint for how to stay ahead. Just as importantly, the community now has evidence that collaborating with the signal (via features) and not solely depending on data can yield a stronger defense.

In sum, our research contributes a significant piece to the puzzle of secure voice authentication. It underscores optimism that with intelligent integration of machine learning and domain expertise, we can build systems capable of detecting and thwarting even the most convincing of audio deepfakes, thereby bolstering trust in voice-based technologies.

REFERENCES

[1] Y. Zhu, C. Goel, S. Koppisetti, T. Tran, A. Kumar, and G. Bharaj, "Learn from Real: Reality Defender's Submission to ASVspoof5 Challenge," 2024.

[2] X. Zhang, J. Yi, C. Wang, C. Y. Zhang, S. Zeng, and J. Tao, "What to Remember: Self-Adaptive Continual Learning for Audio Deepfake Detection," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*, AAAI Press, 2024.

[3] J. Yamagishi *et al.*, "ASVspoof 2021: Accelerating Progress in Spoofed and Deepfake Speech Detection," 2021.

[4] Y. Xu, J. Zhong, S. Zheng, Z. Liu, and B. Li, "SZU-AFS Antispoofing System for the ASVspoof 5 Challenge," 2024.

[5] Y. Xie *et al.*, "Temporal Variability and Multi-Viewed Self-Supervised Representations to Tackle the ASVspoof5 Deepfake Challenge," 2024.

[6] X. Wang, B. Zeng, H. Suo, Y. Wan, and M. Li, "Robust Audio Anti-spoofing Countermeasure with Joint Training of Front-end and Back-end Models," 2023.

[7] X. Wang *et al.*, "ASVspoof 5: Crowdsourced Speech Data, Deepfakes, and Adversarial Attacks at Scale," 2024.

[8] A. Tomilov, A. Svishchev, M. Volkova, A. Chirkovskiy, A. Kondratev, and G. Lavrentyeva, "STC Antispoofing Systems for the ASVspoof2021 Challenge," *Proceedings of ASVspoof 2021*, 2021.

[9] H. Tak, M. Todisco, X. Wang, J. Jung, J. Yamagishi, and N. Evans, "Automatic Speaker Verification Spoofing and Deepfake Detection using Wav2Vec 2.0 and Data Augmentation," *Proceedings of Odyssey 2022: The Speaker and Language Recognition Workshop*, 2022.

[10] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-speaker Conversations Using X-vectors," in *ICASSP 2019*, pp. 5796–5800, doi: 10.1109/ICASSP.2019.8683760.

[11] F. Meriem, B. Messaoud, and Y.-Z. Bahia, "Texture Analysis of Edge-Mapped Audio Spectrogram for Spoofing Attack Detection," *Multimedia Tools and Applications*, vol. 83, pp. 15915–15937, 2024, doi: 10.1007/s11042-023-15329-6.

[12] J. M. Martín-Doñas *et al.*, "ASASVIcomtech: The Vicomtech-UGR Speech Deepfake Detection and SASV Systems for the ASVspoof5 Challenge," 2024.

[13] J. M. Martín-Doñas and A. Álvarez, "The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge," in *ICASSP 2022*, IEEE, 2022.

[14] X. Liu *et al.*, "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2507–2524, 2023, doi: 10.1109/TASLP.2023.3285283.

[15] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition," in *ICASSP 2017*, IEEE.

[16] J. Jung *et al.*, "AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks," in *ICASSP 2022*, IEEE.

[17] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," 2020.

[18] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," *Interspeech*, 2020.

[19] Y. Chen *et al.*, "USTC-KXDIGIT System Description for ASVspoof5 Challenge," 2024.

[20] A. Brown, J. Huh, A. Nagrani, J. S. Chung, and A. Zisserman, "Playing a Part: Speaker Verification at the Movies," in *ICASSP 2021*, IEEE.

[21] K. Borodin *et al.*, "AASIST3: KAN-Enhanced AASIST Speech Deepfake Detection using SSL Features and Additional Regularization for the ASVspoof 2024 Challenge," 2024.

[22] S. Bengio and J. Mariéthoz, "A Statistical Significance Test for Person Authentication," *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2004.

[23] H. Ali, S. Subramani, and H. Malik, "Augmentation through Laundering Attacks for Audio Spoof Detection," 2024.

[24] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *ICASSP 2018*, pp. 5329–5333, doi: 10.1109/ICASSP.2018.8461375.

[25] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," arXiv:1510.08484, 2015, doi: 10.48550/arXiv.1510.08484.

[26] D. S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech 2019*, ISCA, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR Corpus Based on Public Domain Audio Books," in *ICASSP 2015*, pp. 5206–5210, doi: 10.1109/ICASSP.2015.7178964.

[28] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR 2019*, pp. 4685–4694, doi: 10.1109/CVPR.2019.00482.